Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-21М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Бурлаченко В.Ю.
«____» _____ 2022 р.

Керівник проекту
кандидат технічних наук
_____ Буравченко К.О.
«____» _____ 2022 р.
Рецензент _____
_____

Кафедра_КбПЗ_2022рік

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань _12 _____*"Інформаційні технології"*_____

Спеціальність 123 *"Комп'ютерна інженерія"*

Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 6 » вересня 2022 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Бурлаченку Владиславу Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК*

2. Керівник роботи *Буравченко Костянтин Олегович, канд. техн. наук*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 17.08.2022 року

3. Строк подання студентом роботи до захисту *10.12.2022 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*6. Наукова новизна.*
2.Перегляд аналогічних існуючих систем.	*7. Економічна ефективність розробленої програми.*
3. Опис і обґрунтування проектних рішень.	*8. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*9. Висновки.*
5. Впровадження системи в промислову експлуатацію	

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*
Показники економічної ефективності	*1 аркуш*

Кафедра_КібПЗ_2022рік

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	14.11.2022
Охорона праці	Оришака О.В.	06.10.2022	16.11.2022

7. Дата видачі завдання « 6 » вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника
_____ Буравченко К.О.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача
_____ Бурлаченко В.Ю.
(прізвище та ініціали)

Кафедра КбПЗ 2022рік

АНОТАЦІЯ

Бурлаченко В.Ю. Дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи сервісу діагностування та тестування складових ПК.

Метою розробки є дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК.

Об'єктом дослідження є процес сервісу діагностування та тестування складових ПК.

Предметом дослідження є методи сервісу діагностування та тестування складових ПК.

Методи дослідження базуються на методах схемотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи сервісу діагностування та тестування складових ПК.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Embarcadero RAD Studio Delphi 10.3.2 Rio Architect.

Ключові слова: комп'ютерна інженерія, діагностування, тестування

Кафедра_КОПЗ_2022рік

ABSTRACT

Burlachenko V.Yu. Research and software implementation of the service system for diagnosing and testing PC components. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this final qualification work for the second (master's) level of higher education, software was developed, which is intended for the service system of diagnosing and testing PC components.

The purpose of the development is the research and software implementation of the service system for diagnosing and testing PC components.

The object of the study is the service process of diagnosing and testing PC components.

The subject of the study is the methods of the service of diagnosing and testing PC components.

Research methods are based on circuit engineering methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the service system for diagnosing and testing PC components.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Embarcadero RAD Studio Delphi 10.3.2 Rio Architect environment.

Keywords: computer engineering, diagnostics, testing

Кафедра_КОПЗ_2022рік

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 3
ВСТУП..... 4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 7
1.1 Призначення системи..... 7
1.2 Область застосування..... 8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... 21
2.3 Розгорнута постановка завдання 25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 27
3.1 Опис функціонування системи 27
3.2 Розробка структурної схеми..... 41
3.3 Розробка функціональної схеми 42
3.4 Розробка діаграми процесів..... 43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... 45
4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... 45
4.2 Захист розробленого програмного забезпечення..... 56
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 58
6 НАУКОВА НОВИЗНА 66

Кафедра_КОПЗ_2022рік

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ..... 67
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 67
7.2 Розрахунок трудомісткості розробки програмної продукції..... 69
7.3 Визначення чисельності виконавців і планового фонду зарплати..... 71
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника..... 76
7.5 Визначення собівартості розробки та ціни програмної продукції..... 80
7.6 Визначення об’єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції..... 83
7.7 Визначення експлуатаційних витрат..... 83
7.8 Визначення економічної ефективності програмної продукції..... 85
7.9 Висновок..... 87
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ..... 88
8.1 Вступ..... 88
8.2 Аналіз умов праці на робочому місці ІТ-фахівця..... 89
8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців..... 91
8.4 Розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці..... 93
8.5 Висновки до розділу..... 96
9 ОСНОВНІ ВИСНОВКИ..... 98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 100

КафеДра_КОПЗ_2022рік

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ОС	–	операційна система
ПЗ	–	програмне забезпечення
ПЗП	–	постійний запам’ятовуючий пристрій
AGP	–	Accelerated Graphics Port
API	–	Application Programming Interface
BIOS	–	Basic Input-Output System
CD	–	компакт-диск
COM	–	послідовний порт
DMA	–	прямий доступ до пам’яті
DVD	–	Digital Versatile Disc
GPU	–	специфічні блоки графічного процесора
I/O	–	Ввод/Вивід
ISA	–	Industry Standard Architecture
PCI	–	Peripheral component interconnect
IRQ	–	пріоритет апаратних переривань
LPT	–	паралельний порт
SCSI	–	Small Computer System Interface
S.M.A.R.T.	–	Self Monitoring Analysing and Reporting Technology
USB	–	універсальна серійна шина
VCL	–	бібліотека візуальних компонентів

Кафедра_КОПЗ_2022рік

ВСТУП

Актуальність теми. На перший погляд більшості домашніх користувачів зовсім не потрібно розбиратися в залізній начинці свого комп'ютера – але лише за умови, якщо хто-небудь інший візьме на себе рішення питань, пов'язаних з «залізом». А якщо таких не найдеться, то ознайомитися з деякою початковою інформацією із цієї області зовсім не зашкодить. Справа в тому, що рано або пізно вам доведеться проводити апгрейд вашого ПК, що далеко не завжди припускає банальну зміну одного системного блоку з начинкою на інший, більш сучасний. Можливо, у багатьох випадках виявиться досить замінити (або доповнити) його окремі компоненти, наприклад просто докупити пам'ять. Але, щоб вибрати вірну стратегію апгрейду, варто чітко знати, яке апаратне забезпечення у вас установлено, і зрозуміти, чого конкретно не вистачає вашому комп'ютеру для більш швидкої роботи: потужності процесора, пам'яті, швидкості вінчестера й т.п. Але це лише одна із проблем. Інша, не менш важлива пов'язана з тим, що любе встаткування в якийсь момент починає працювати зі збоями або взагалі виходить із ладу. Сумні наслідки подібних ситуацій очевидні, тому набагато надійніше контролювати стан важливих апаратних систем, що дозволить із певною ймовірністю прогнозувати можливе поводження апаратури в найближчому майбутньому, вчасно виявити ті або інші неполадки з комп'ютерними компонентами й вчасно подбати про запобіжні заходи. І нарешті, припустимо, що ви придбали новий комп'ютер або провели апгрейд старого – очевидно, що вам необхідно швидко зорієнтуватися й зрозуміти, чи дійсно комп'ютер стабільно працює й чи відповідає начинка його системного блоку заявленій при покупці, причому не розбираючи сам блок, на якому може стояти пломба.

Кафедра_КОПЗ_2022рік

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем сервісу діагностування та тестування складових ПК.

– Дослідження системи сервісу діагностування та тестування складових ПК.

– Програмна реалізація системи сервісу діагностування та тестування складових ПК.

Об'єктом дослідження є процес сервісу діагностування та тестування складових ПК.

Предметом дослідження є методи сервісу діагностування та тестування складових ПК.

Методи дослідження базуються на методах схемотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод сервісу діагностування та тестування складових ПК.

– Розроблено вітчизняний продукт сервісу діагностування та тестування складових ПК, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі сервісу діагностування та тестування складових ПК.

Кафедра_КОПЗ_2022рік

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVI Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2022, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №13.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

Кафедра_КбПЗ_2022рік

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Будь-який користувач повинен мати прості інформаційно-діагностичні інструменти, які надавали б інформацію про наявне апаратне забезпечення (процесорі, відеоадаптері, портах, принтерах, звуковій карті, пам'яті й т.д.), контролювали життєво важливі параметри найбільш критичних з погляду виходу з ладу апаратних компонентів і дозволяли оцінити продуктивність і стабільність роботи комп'ютера. Таким чином програмний продукт, який розробляється у даному магістерському проекті призначений для діагностування та тестування пристроїв ЕОМ. Він повинен реалізовувати наступні функції:

– Видавати загальну інформацію, тобто тип процесора, операційна система, версія ОС, ім'я користувача, організація.

– Видавати інформацію про всі диски, установлені у системі, таких як жорсткі диски, USB-flash диски, CD/DVD-rom, а також інформацію про їхню ємність, мітку тому, і файловій системі.

– Видавати інформацію про принтери, установлені у системі.

– Видавати інформацію про системну пам'ять, про її кількість, інформацію про файл підкачки.

– Видавати інформацію про встановлені в системі клавіатурі й миші.

– Видавати інформація про відеокарту, розрішення монітора, постачальника драйверу й т.д..

– Видавати інформацію про процесор – вендор, частота.

– Видавати інформацію про встановлені шрифти.

– Проводити діагностику – такі властивості, як тестування процесора, пам'яті й перевірка лотка CD/DVD-rom.

– Видавати інформацію про автора цієї програми.

Кафедра_КОПЗ_2022рік

1.2 Область застосування

Існує множина інших програм діагностики комп'ютера, і у Вас може виникнути цілком природне запитання, а в якому ступені моя утиліта краще існуючих, або хоча б чим вона виділяється з їхнього загального ряду? Кращу відповідь Ви одержите, якщо я опишу Вам одну конкретно взяту програму, що вважається однією з найбільш «просунутих» у своєму класі – це Sisoft Sandra (System Analyzer Diagnostic and Reporting Assistant).

До складу повної версії пакета входять близько 70 модулів для збору інформації про всі основні компоненти PC (включаючи CPU, вінчестер, відео– і аудіосистеми, принтери, комунікаційне встаткування, клавіатура, миша та ін.), тестування (перевіряються CPU, дискові накопичувачі, пам'ять і мережеве встаткування) і вироблення рекомендацій з поліпшення їхньої роботи. Є можливість перевірки розташування й умісту основних конфігураційних файлів. Графічний інтерфейс програми досить наочний і дозволяє одержати саму повну інформацію про комп'ютер, включаючи часом і недокументовану.

Єдиний недолік Сандри – це те, що її професійна версія коштує 29 доларів.

Виходячи з вищевказаного, я дійшов висновку, що єдиними перевагами моєї утиліти є її безкоштовність для кінцевого користувача й те, що вона є вітчизняним продуктом.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

Кафедра_КПЗ_2022рік

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

У рамках завдань визначених у магістерській роботі, мною проведений огляд існуючих програм діагностики комп'ютера.

Були розглянуті наступні програми:

– SiSoftware Sandra;
– Fix-It Utilites;
– 3DMark;
– ASTRA64 – Advanced System Information Tool;
– Hard Drive Inspector;
– EVEREST Ultimate Edition;
– BurnInTest.

SiSoftware Sandra

SiSoftware Sandra – по моєму краща програма яка сполучає в собі дві категорії: тестування та діагностику. Сторінка програми перебуває за адресою www.sisoftware.net. Вся інформація виводиться у вигляді модулів:

– Information modules – найдокладніша інформація про систему, компоненти комп'ютера, системних процесах, запущених сервісах, а також докладна інформація про Direct і OpenGL.

– Benchmarking modules – вимір продуктивності процесора, оперативної й кеш пам'яті, файлової системи жорстких дисків і CD/ DVD-rom.

– Listing modules – тут можна переглянути зміст системних файлів: boot.ini, system.ini, win.ini, autoexec.bat, config.sys, а також найважливіші значення реєстру й типи файлів на даному комп'ютері.

Кафедра_КбПЗ_2022рік

– Testing modules – інформація об переривання IRQ, установки DMA, I/O портів вводу-виводу, ресурси системної пам'яті й настроювання Plug&Play.

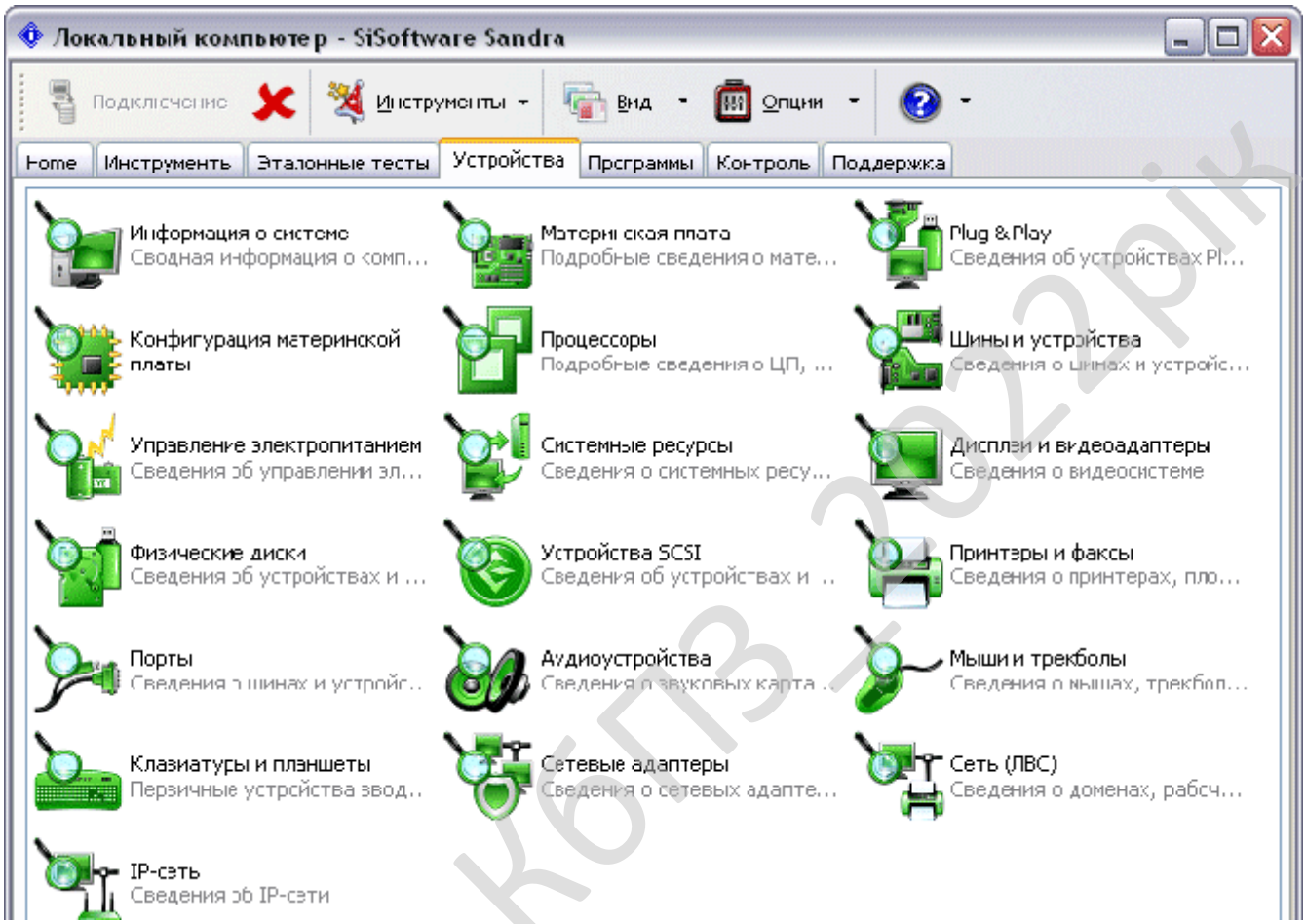

Рисунок 2.1 – Основні модулі програми SiSoft Sandra

Користуватися програмою простіше простого, вибираєте потрібний модуль і одержуєте докладну інформацію. Якщо який-небудь компонент не підтримує деякі функції, програма це позначає червоним хрестиком. Знайдені помилки й те, на що варто звернути особливу увагу, позначаються жовтогарячим трикутником зі знаком оклику усередині. Навіть у нерусифікованій версії програми розібратися дуже легко. Окремо варто виділити модуль benchmark, що дозволяє вимірювати продуктивність процесора, пам'яті й файлової системи в

Кафедра_КОПЗ_2022рік

порівнянні з іншими конфігураціями комп'ютерів. Просто вибираєте потрібний тест, і через пару хвилин можете порівнювати свій результат з іншими.

Fix-It Utilites

Fix-It Utilites – цілий пакет корисних утиліт, при бажанні з якими ви розберетеся самостійно. Зараз же нас цікавить вкладка – system diagnostics**,** далі симпатична кнопка **–** PCDiagnostic при натисканні на яку, комп'ютер недовго задумається, і через кілька секунд виведе на екран вікно вітання й список знайдених пристроїв. Тут головне, правильно вибрати те, що вас цікавить, інакше, завдяки декільком цікавим тестам, процес може затягтися на кілька годин. За допомогою цієї програми я не раз виловлював неробочі LPT, COM і USB порти. Бувало й непрацюючу оперативну й відео пам'ять, нестабільно працюючі жорсткі диски й CD/DVD-rom. Також у програмі є прості й зручні тести клавіатури, модему, аудіопристроїв, мережевої плати, монітора й мультимедійних функцій операційної системи. Не пройдені тести позначаються червоним, праворуч виводиться докладна інформація про неполадку і її можливі причини.

Fix-It Utilities містить у собі антивірус/антишпигун, утиліти для діагностики й захисту системи, утиліти для резервного копіювання на випадок краху системи й багато чого іншого.

Програма може виконувати й роль системного твікера, дозволяючи без труда настроїти систему для більш швидкої й безпомилкової роботи.

Велика кількість різних утиліт, що становлять пакет, згруповані в три основних роздягнула:

1.One-Click Solutions – швидке чищення від сміття дисків і реєстру.

2. Protection – антивірус і антишпигун.

3. Maintenance (обслуговування):

– System Diagnostics – інструментальні засоби діагностики системи.

– Fix & Maintain – комплект утиліт для виправлення помилок у системі.

– System Optimize – комплект утиліт для оптимізації системи.

Кафедра_КбПЗ_2022рік

– System Cleanup – комплект утиліт для чищення від сміття дисків і реєстру.

– System Recovery – утиліти й сервіси для відновлення даних після краху системи.

Рисунок 2.2 – Діагностичні тести в програмі Fix-It Utilites

Сайт програми www.v-com.com . Рекомендую встановити цей пакет на свій комп'ютер, тому що крім діагностики заліза, у нього входить купа інших відмінних утиліт. Наприклад DiskFixer – альтернатива вбудованому в Windows ScanDisk, JetDefrag – альтернативна утиліта для дефрагментації файлової системи, FileUndeleter – відновлення вилучених файлів, DiskCleaner – чистильник диска, MediaVerifier – перевірка читаності файлів на компакт дисках і флопі

Кафедра_КПЗ_2022рік

дискетах. А також спеціальні утиліти для збереження й відновлення критично важливих системних файлів, образа диска, і окремий розділ SystemRegistry для пошуку помилок, дефрагментації, чищення й зручного редагування реєстру.

3DMark

3DMark – назва серії широко відомих комп'ютерних бенчмарків, розроблених фінською компанією Futuremark (раніше MadOnion.com) (сайт Сайт www.madonion.com). Бенчмарки даної серії орієнтовані на тестування насамперед графічного компонента персонального комп'ютера з метою визначення продуктивності системи в комп'ютерних іграх. Основне призначення 3DMark – тестування продуктивності й стабільності графічної плати (відеокарти) і оцінка її продуктивності в умовних одиницях. Останні версії 3DMark, крім відеокарти, тестують також продуктивність центрального процесора в таких задачах, як ігровий штучний інтелект і фізичний движок. 3DMark, по суті, візуально являє собою комп'ютерну гру, що є неінтерактивної, тому що користувач не може впливати на геймплей. Бенчмарки 3DMark є пропрієтарними комерційними програмами, однак всі бенчмарки серії, за винятком останньої (3DMark Vantage), мають урізані безкоштовні версії з обмеженою функціональністю.

3DMark позиціонується розроблювачами як об'єктивний, незалежний, нейтральний і достовірний засіб для тестування продуктивності й стабільності комп'ютерних систем. Більше того, розроблювачі позиціонують 3DMark, особливо її ігрові тести, як майбутнє комп'ютерних ігор. Так, у всіх ігрових тестах присутні графічні технології, які перебувають на етапі доробки й впровадження в комп'ютерні ігри, а іноді й зовсім відсутні. Споконвічно серія 3DMark орієнтувалася на Direct API і використовувала тільки його.

У більшості випусків 3DMark всі тести можна розділити на дві групи: ігрові тести й синтетичні специфічні тести. Перші виявляють собою неінтерактивну майже повноцінну комп'ютерну гру, що працює в режимі реального часу з використанням ігрового движка. На відміну від повноцінної комп'ютерної гри користувач не може впливати на геймплей і управляти

Кафедра_КОПЗ_2022рік

протагоністом або віртуальною камерою, він може лише спостерігати. Під час тесту заміриться кількість кадрів і середня частота кадрів у секунду. Інший тип тестів завантажує обчисленнями й оцінює лише конкретні специфічні блоки графічного процесора (GPU), наприклад: шейдерні блоки, блоки текстурування, растеризації й т.д. Хоча ці синтетичні тести не відбивають продуктивність відеокарти в іграх, однак вони дозволяють досить точно оцінити продуктивність конкретних блоків GPU і на цій підставі дати більше об'єктивну оцінку продуктивності GPU.

Існує декілька зовсім різних версій програми, але всі вони орієнтовані винятково на вимір продуктивності відеосистеми. Використовувати останню версію для комп'ютера із застарілою відеокартою не має змісту, програма просто відмовиться запускатися або покаже всього пари тестів у вигляді слайд-шоу. Думаю, версія 2001 підійде для більшості, колись популярних конфігурацій комп'ютерів з відеокартами Riva 16-64 і JeForce 64-64Mb.

При запуску програми, у головному вікні ви побачите тільки модель відеокарти й характеристики монітора. Для одержання більш докладної інформації, тисніть на SystemInfo, там можна довідатися – модель процесора, розмір кеш пам'яті, версію direct і іншу системну інформацію. У програмі можливий вибір всіх або тільки деяких тестів. Майже всі тести проводяться два рази, на низкою й високої деталізації, що дає більшу точність. Приголомшлива графіка зробила цю програму популярної, серед аматорів комп'ютерних ігор, і бажаючих похвастати продуктивністю свого ПК перед друзями. Після тесту програма виводить результат у вигляді балів, які можна зрівняти з іншим комп'ютером. Головне, що тест відеосистеми, не обходиться без критичного навантаження на інші компоненти комп'ютера. І якщо тестуємий комп'ютер з ними впорався, значить швидше за все, основні компоненти в порядку.

Кафедра_КОПЗ_2022рік

Рисунок 2.3 – Програма для тестування швидкості роботи відеосистеми

ASTRA64 – Advanced System Information Tool 2.00

Програма визначення конфігурації й діагностики комп'ютера. Дозволяє одержати докладну інформацію (у тому числі недокументовану) про апаратне забезпечення комп'ютера й режимах його роботи:

– Містить унікальну функцію пошуку пристроїв з невстановленими драйверами й функцію швидкої перевірки надійності вінчестерів.

– Дозволяє одержати відомості про процесор, материнську плату, тверді диски, S.M.A.R.T., CD/DVD, SCSI пристроях, модулях пам'яті, чипсеті, BIOS,

Кафедра_КБП_2022рік

PCI/AGP, USB і ISA/Pn пристроях, моніторі, відеокарті, звуковій і мережевій карті, принтері, установлених програмах і відновленнях й про багато іншого.

– Створення файлу-звіту в текстовому, INI, HTML, XML і CSV форматах, можливість експорту даних у програми обліку обчислювальної техніки.

– Можливість роботи в режимі командного рядка.

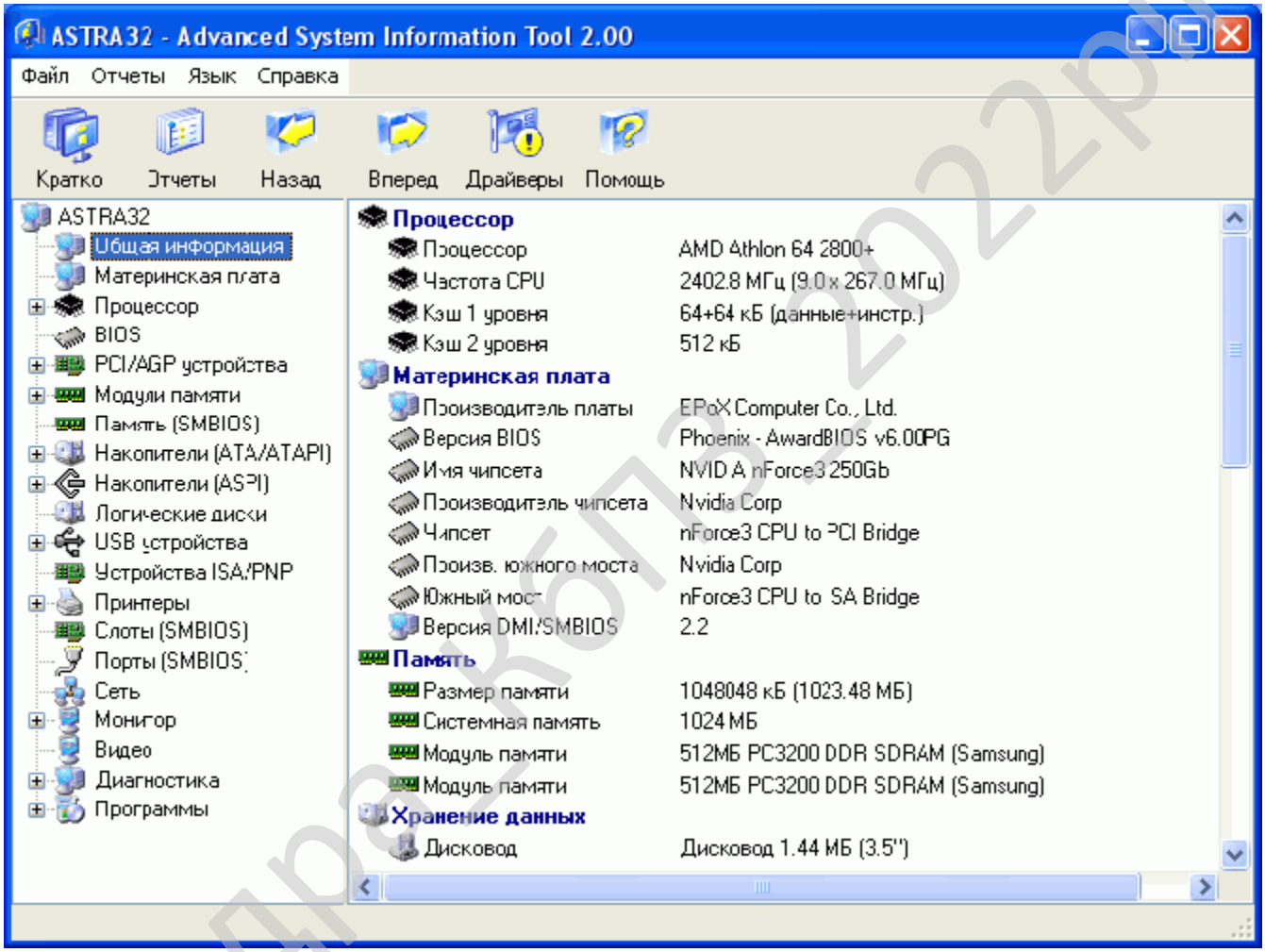

Рисунок 2.4 – Скріншот програми ASTRA64

Hard Drive Inspector

Hard Drive Inspector – потужна, зручна й ефективна програма, призначена для контролю за станом жорстких дисків. Цінність даних, збережених на жорсткому диску комп'ютера, найчастіше перевершує вартість комп'ютера як

Кафедра КОПЗ 2022рік

такого. Однак жорсткий диск – це одна із самих ненадійних комп'ютерних компонентів. Коли він виходить із ладу (що трапляється досить часто), інформація, збережена на ньому, губиться частково або повністю.

Сучасні жорсткі диски підтримують технологію S.M.A.R.T. , що була спеціально розроблена для виявлення прийдешніх виходів дисків з ладу. Використовуючи цю технологію, Hard Drive Inspector може здійснювати спостереження за поточним станом різних життєво важливих параметрів, що описують надійність і продуктивність диска. Hard Drive Inspector аналізує ці дані й попереджає користувача в тому випадку, коли параметри вказують на критичне падіння надійності вінчестера. Це в багатьох випадках дозволяє запобігти втраті інформації.

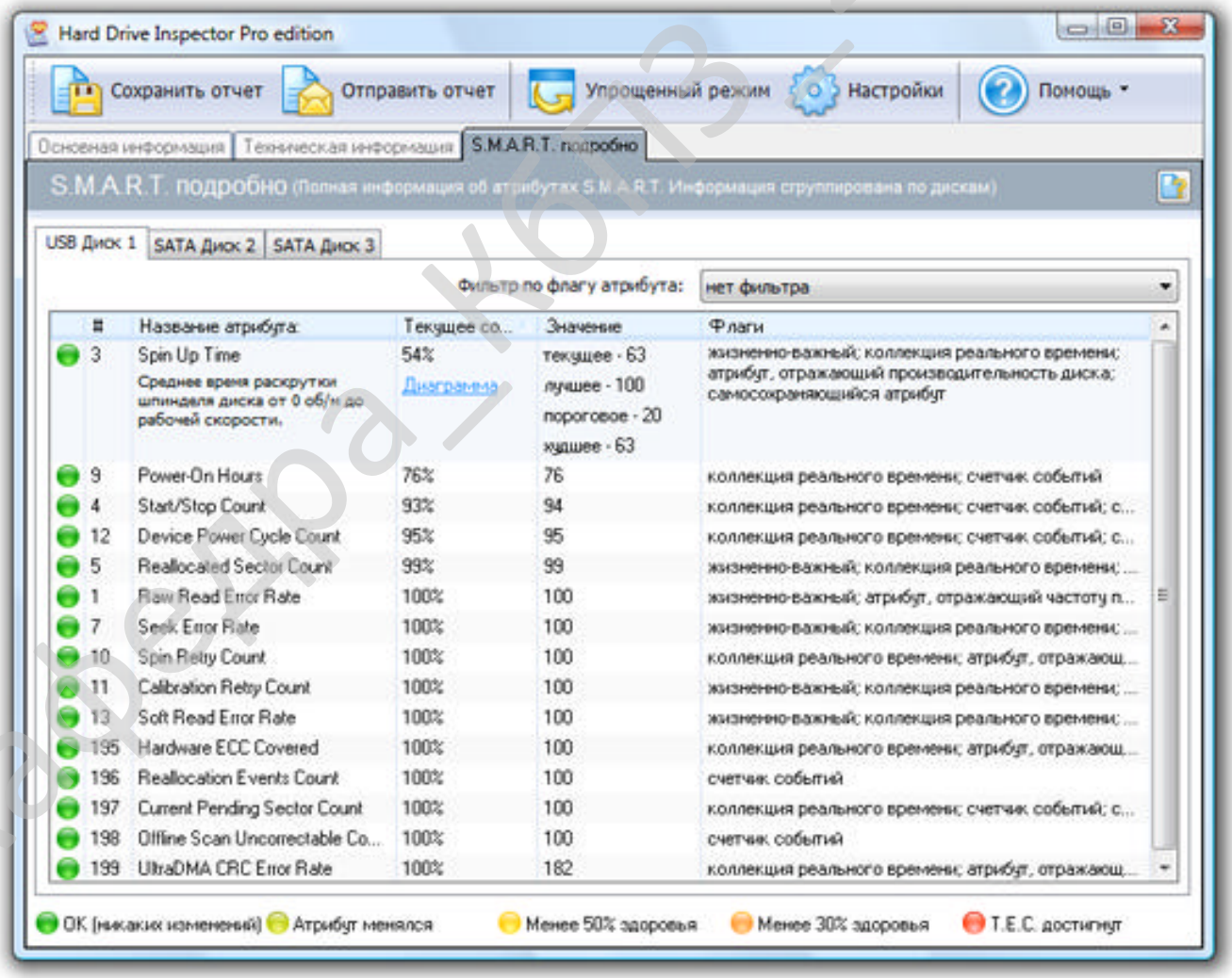

Рисунок 2.5 – Скріншот програми Hard Drive Inspector

Кафедра КБПЗ 2022рік

EVEREST Ultimate Edition

EVEREST Ultimate Edition – багатофункціональна утиліта для діагностики системи й бенчмаркинга (оцінки ефективності), призначена для просунутих користувачів комп'ютера. Дане рішення призначене для оптимізації й тестування системи, моніторингу апаратного забезпечення й діагностики нововведень і визначення потреб системи.

Тестування CPU, FPU і Memory BenchMark дозволяє знайти найбільш ефективне рішення для роботи системи, а також – зрівняти поточний стан системи з попередніми варіантами або з іншими системами. Вигідна відмінність програми EVEREST Ultimate Edition від інших аналогічних продуктів – надання повної інформації щодо встановленого програмного забезпечення, оперативної системи й ступеня безпеки й повна діагностика системи, здатна зібрати більше 100 сторінок інформації про систему.

Основні можливості:

1. Інформація про апаратне забезпечення. Повна інформація про складових комп'ютерів: материнських платах, процесорах, відео– і мережевих адаптерах, дисководах і пристроях уведення, мультимедіа, а також портах, зовнішніх підключених пристроях і керуванні харчуванням.

2. Інформація про програмне забезпечення. Детальна інформація про мережеві підключення й ресурси, користувачах, групах, поштових облікових записах, установках мережі, програмному забезпеченні й операційній системі в цілому.

3. Безпека. Запобігання виконання даних (наприклад, шкідливого коду) за допомогою функції DEP (Data Execution Prevention), спеціальні додатки для створення файрволу, антишпигунів, антитроянів і антивірусів.

4. Діагностика

– Інтерфейс оснащений вбудованою панеллю EVEREST CPUID, на якій компактно відображається інформація про процесор, материнську плату, оперативну пам'ять і архітектурі комп'ютера.

Кафедра_КПЗ_2022рік

– Також надаються дані про можливості «розгону» системи, динамічні відновлення й підтримка технологій Enhanced Intel SpeedStep і AMD Cool'n'Quiet.

– Діагностика програмного забезпечення дозволяє відслідковувати температуру процесорів, стан вентилятора й відносне зношування диска.

– Відстеження проблем сумісності.

5. Звіти

Підтримується 7 різних форматів звіту, з можливістю роздруківки й відправлення по електронній пошті.

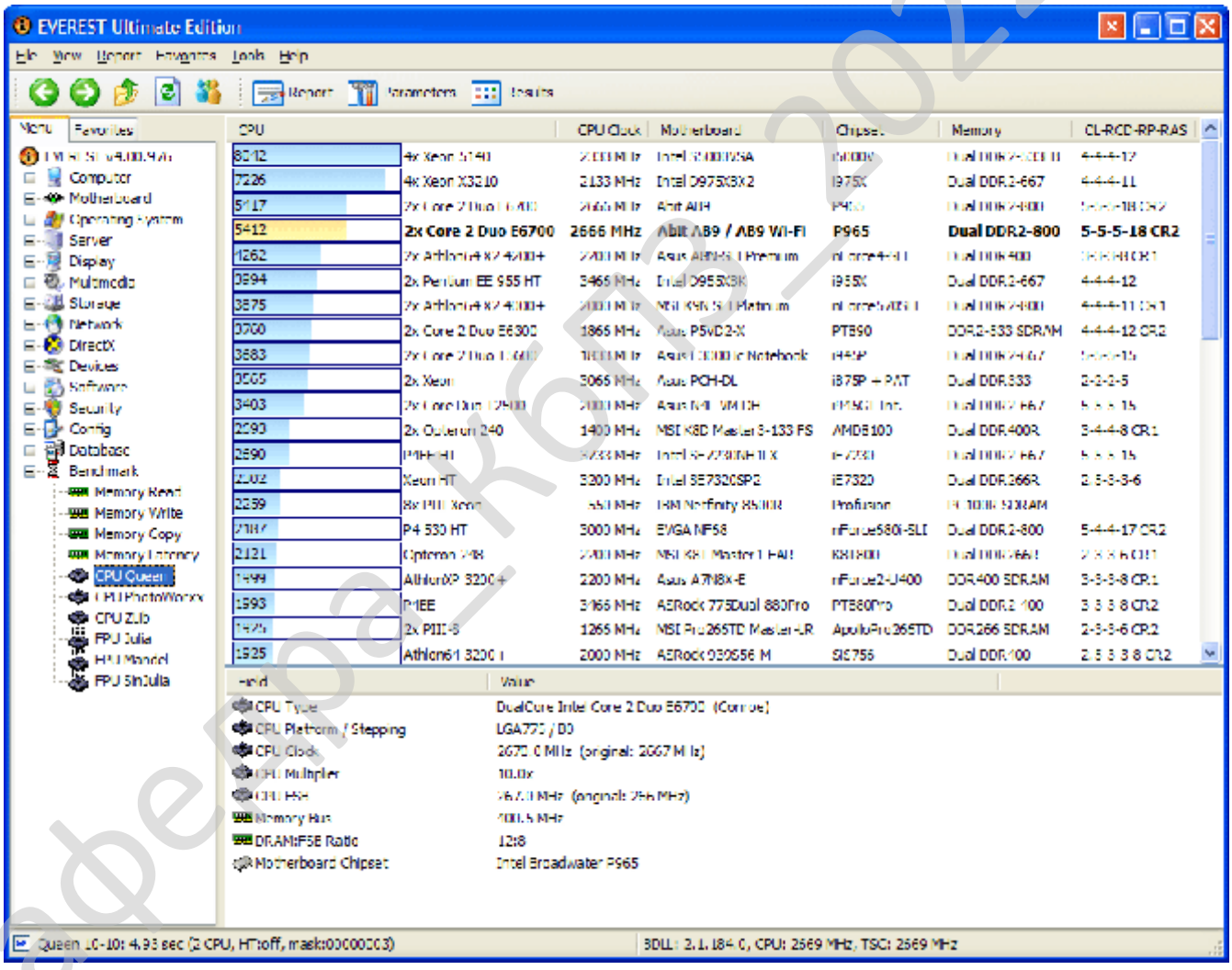

Рисунок 2.6 – Скріншот програми EVEREST Ultimate Edition

Кафедра КБТС 2022рік

BurnInTest

Набір утиліт для перевірки стабільності роботи комп'ютера в режимі максимального навантаження. Професійна версія програма для всебічного й детального тестування компонентів комп'ютера. Утиліта тестує швидкість процесора, оперативну пам'ять, жорсткі диски, CD/DVD приводи, звукові карти, відеокарти, принтери, мережеві з'єднання. BurnInTest Professional також може проводити тестування послідовних (до 12 портів одночасно) і паралельних портів, портів USB (до 10 портів одночасно). У професійній версії є додаткові параметри тестування процесорів і відеокарт.Після тестування програма видає звіт зі знайденими помилками й нестабільностями. BurnInTest Pro ідеально підходить для аматорів розгону

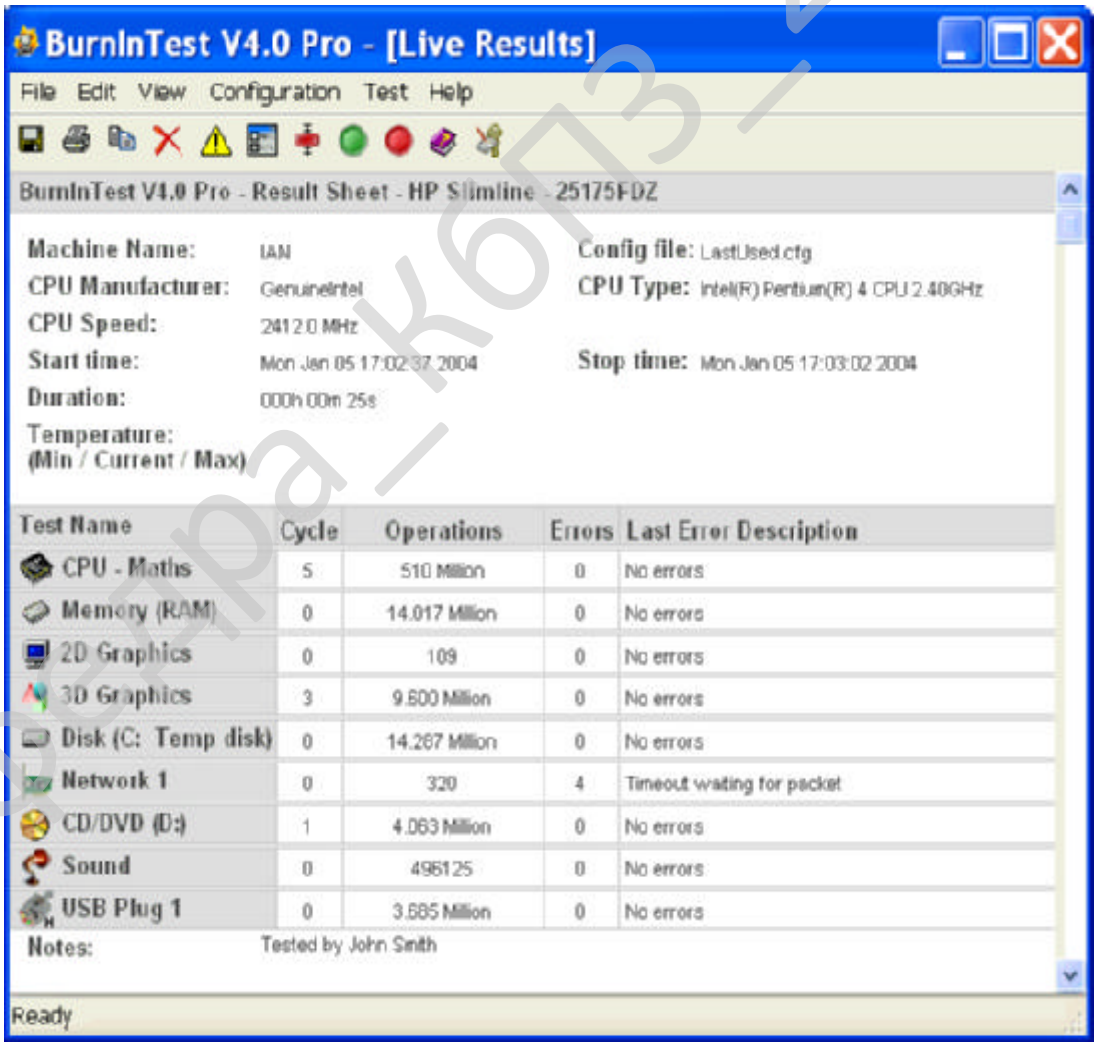

Рисунок 2.7 – Скріншот програми BurnInTest

Кафедра_КбПЗ_2022рік

Звичайно, розглянутими вище утилітами все різноманіття програм, що відображають "залізну" начинку комп'ютера, не обмежується, але навряд чи ті, що залишилися за рамками огляду, хоч у чомусь принципово відрізняються від уже розглянутих. По великому рахунку, кожна з них здатна дати практично вичерпну інформацію про апаратну конфігурацію комп'ютера.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero RAD Studio Delphi 10.3.2 Rio Architect – це найшвидший спосіб створювати й обновляти інтенсивно працюючі з даними, сильно взаємодіючі застосунки з візуально насиченим користувальницьким інтерфейсом для Windows 10, Mac, мобільних пристроїв, IoT і інших платформ за допомогою Object Pascal і C++. Широкий вибір функцій підтримки Windows 10, у тому числі нові компоненти VCL для Windows 10, стилі для VCL і FMX, а також служби UWP (універсальної платформи Windows), наприклад повідомлення, дозволяють легко й швидко перенести застосунки в Windows 10, зберігши користувачів. Нова платформа дозволяє підтримувати великі проекти на більшому числі платформ із подвоєним обсягом пам'яті в середовищі розробки й удвічі більшим розміром підтримуваних проектів. Крім того, підтримка декількох моніторів і десятки нових функцій середовища розробки, призначених для прискорення створення коду, зроблять роботу як ніколи ефективною. За допомогою RAD Studio 10 розроблювачі зможуть створювати застосунки в 5 разів швидше в порівнянні з іншими інструментами, а розробка застосунків для декількох настільних, мобільних, хмарних платформ і платформ баз даних, включаючи 64і 64-розрядні версії Windows 10, Mac OS X, iOS і Android, стане ще швидше.

Зміни у версії 10.3 Rio:

– Створюйте міжплатформені застосунки. 80% інтернет користувачів мають смартфони й застосунки доступу, а також дані з мобільного пристрою й

Кафедра_КОПЗ_2022рік

ноутбука / настільного комп'ютера, саме тому так важливо в цей час, щоб застосунки працювали в будь-якому пристрої.

– У всіх версіях Professional, Enterprise і Architect RAD Studio 10.3 надається підтримка процесу розробки застосунків для мобільних пристроїв. Розроблювачі RAD Studio кодують лише один раз, компілюють споконвічно для кожної платформи, що скорочує час і трудозатрати на вивчення декількох мов і дозволяє паралельно управляти циклами розробки.

– Підтримка Android API26, відповідність вимогам Google Play Store відносно нових застосунків із серпня 2018 року й відновлення застосунків з листопада 2018 року.

– Власні елементи керування Android і стилізовані елементи керування FMX в одній і тій же формі Android, включаючи тему матеріального дизайну для Android 5.0 або вище.

– Підтримка iOS 12 (64і 64-біт) для створення App Store і корпоративних застосунків.

– Підтримка смайликів Юнікод.

– Програмуйте по-своєму. Завдяки двом новим темам самостійне налаштування інтегрованого середовища розробки для відповідності вашому стилю кодування ще ніколи не була настільки простій.

– Темне й світле оформлення Незалежно від того чи волієте ви кодувати вночі або у світлий час доби, завдяки темному й світлому оформленню RAD Studio ви можете вибрати потрібний вам стиль. Було доведено, що темне оформлення допомагає знизити зорову напругу в умовах низького освітлення, дозволяючи вам працювати більш продуктивно вночі. Немає нічого простіше, ніж перейти від темного до світлого оформлення й навпаки за допомогою меню панелі інструментів.

– Виконаєте користувальницьке налаштування свого середовища розробки Поліпшена програма установки інтерфейсу користувача й менеджера ліцензій інтерфейсу користувача дозволяє визначити ті можливості, які необхідні

Кафедра_КбПЗ_2022рік

й опустити непотрібні, незалежно від того чи розробляєте ви застосунки для декількох платформ або всього однієї.

– Чистий, оновлений інтерфейс користувача інтегрованого середовища розробки Знайдіть потрібні можливості. Швидко. Головне вікно інтегрованого середовища розробки відцентровано й відрізняється високим ступенем читаності. Ви з легкістю визначите, де перебуває область фокусування клавіатури з оновленими змінами фонових квітів фокуса. Вкладки редактора більше, що полегшує читання шрифтів, тому ви можете швидко внести зміни й зберегти кодування.

– Чудові застосунки Windows з VCL. Бібліотека візуальних компонентів (Visual Component Library, VCL) пропонує просту й візуальну розробку користувальницького інтерфейсу застосунки, у версії 10.3 представлені нові відновлення, які дозволять вашим додаткам виглядати сучасними й свіжими.

– Розширена підтримка HighDPI. Завдяки новому елементу керування VCL High DPI ImageList у версії 10.3 розроблювачі, що створюють нові застосунки VCL для Windows або оновлюючі існуючі застосунки для High DPI дисплеїв, можуть повністю підтримувати зроблені до рівня пікселів зображення зі змінною розв’язною здатністю на всіх елементах керування, а також будь-яке користувальницьке креслення, що вимагає масштабованих зображень для моніторів з різною розв’язною здатністю.

– Підтримка Per Monitor V2. Переконаєтеся, що ваш додаток масштабується правильно для всіх типів масштабування в Windows, реагуючи на зміни масштабування DPI на різних екранах під час виконання.

– Розширена підтримка Windows 10 і WinRT API. Сюди відноситься ряд ключових API-інтерфейсів WinRT і останні API-інтерфейси Windows 10, включаючи готові до використання компонента для убудованих у застосунки покупок і випробувань у магазині Windows 10 Store.

Кафедра_КоПЗ_2022рік

– Розгортання застосунків на основі служб за допомогою RAD Server. Продуктивність RAD Server була значно поліпшена завдяки десятикратному збільшенню потужності відносно простих операцій.

– Нові компоненти обробки JSON допоміжного засобу.

– Розширена підтримка RAD Server для клієнта Ext JS. Об'єднайте зовнішній інтерфейс javascript і веб-службу, підтримувану REST Server REST. (У версії Architect тепер включена ліцензія ExtJS Professional).

– Версії Enterprise включають ліцензію для одиничного розгортання RAD Server.

– Версії Architect включають ліцензію для розподіленого розгортання RAD Server.

– Що нового в C++? Підтримка C++17 Win64 збільшує продуктивність, поліпшує роботу компілятора й прискорює процес кодування. Були оновлені RTL і STL.

– Нова версія STL/Dinkumware 2018 для Win64 і Win64.

– Поліпшене автодоповнення коду Автодоповнення коду для даного компілятора тепер асинхронне, швидше й із кращими результатами, чим у попередньому автодоповненні коду C++. Уведення тексту не буде припинятися, поки виконується розрахунок.

– Тепер є підтримка налагодження для оптимізації компонувань.

– 2X швидкість математичної продуктивності для Win64.

– Нові додаткові лабораторії C++ в GetIt.

– Нові й поліпшені можливості роботи з базами даних. InterBase 2017 / IBToGo 2017 в RAD Studio. Версії Professional включають ліцензію розроблювача InterBase 2017, у той час як версії Enterprise і Architect містять у собі ліцензії InterBase ToGo. InterBase ToGo доповнена можливістю шифрування, функціями зміни подань, призначених для простої синхронізації даних застосунку по підписці без обмежень за розміром файлу бази даних.

Кафедра_КоПЗ_2022рік

– Поліпшена й оновлена підтримка для популярних баз даних, включаючи mySQL v8.0, MariaDB 10.3, SQL Server 2017, PostgreSQL v10, Firebird v3.0, MongoDB, InterBase, SQLite 3.23.1, SQL Anywhere і багатьох інших.

– Удосконалення DataSnap.

– Поліпшення REST. Підтримка додаткових родинних REST методів, типів і властивостей.

– Повністю оновлений модуль живлення версії Architect. Одержіть більше від версії Architect, включаючи ці ліцензії сімейства Idera.

– Ліцензія Sencha ExtJS Professional: Створіть свій ідеальний мережевий вхідний інтерфейс за допомогою javascript і ExtJS.

– Ліцензія на розгортання InterBase ToGo. Додайте сховище даних у свої застосунки за допомогою цієї гнучкої, зашифрованої бази даних, що вбудовується.

– Ліцензія для розподіленого розгортання RAD Server. Ідеально підходить для серверного застосунку архітектури мікросервісів.

– Ліцензія AquaData Studio. Вражаючий аналіз бази даних.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи сервісу діагностування та тестування складових ПК.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

Кафедра_КОПЗ_2022рік

Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

Кафедра_КбПЗ_2022рік

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Опис Win64 API й об'єктів VCL

Задача поставлена в магістерському проекті вирішувалася за допомогою функцій Win64 API й об'єктів VCL. Тому в цьому розділі дамо опис даних технологій.

Windows API – найбільш важлива й потужна додаткова бібліотека функцій, доступна кожному Delphi-програмістові. Багато хто з них, у тому числі й досвідчених розроблювачах працюють із ними, використовуючи прості готові рішення, почерпнуті в різних книгах і журналах, не дуже замислюючись про суть цієї технології. Такий підхід є достатнім при рішенні простих задач, але для серйозної роботи переважніше більш детально розібратися з основними принципами використання функцій Windows API. Чим ми зараз і займемося.

Windows API – набір функцій операційної системи

Абревіатура API для багатьох починаючих програмістів виглядає досить таємниче. Насправді ж Application Programming Interface (API) – це просто деякий готовий набір функцій, що можуть використовувати розроблювачі додатків. У загальному випадку дане поняття еквівалентно тому, що раніше частіше називали бібліотекою підпрограм. Однак найчастіше під API мається на увазі деяка особлива категорія таких бібліотек.

У ході розробки практично будь-якого досить складного додатка (MyAppication) для кінцевого користувача формується набір специфічних внутрішніх функцій, використовуваних для реалізації даної конкретної програми, що називається MyApplication API. Часто виявляється, що ці функції можуть ефективно використовуватися також для створення інших додатків, у тому числі іншими програмістами. У цьому випадку автори виходячи зі стратегії просування

Кафедра_КОПЗ_2022рік

свого продукту повинні вирішити питання – чи відкривають вони доступ до цього набору для зовнішніх чи користувачів ні? При позитивній відповіді на нього в описі програмного пакета, як його переваго, з'являється фраза про те, що "комплект включає відкритий набір API-функцій" (але іноді за додаткові гроші).

Таким чином, найчастіше під API мається на увазі набір функцій, що є частиною одного додатка, але при цьому доступних для використання в інших програмах. Наприклад, Excel крім інтерфейсу для кінцевого користувача має набір функцій Excel API, що може використовуватися, зокрема, при створенні додатків за допомогою Delphi.

Відповідно, Windows API – це набір функцій, що є частиною самої операційної системи й у той же час – доступної для будь-якого іншого додатка, у тому числі написаного за допомогою Delphi. І в цьому плані цілком виправдана аналогія з набором системних переривань BIOS/DOS, що фактично являє собою DOS API

Відмінність полягає в тім, що склад функцій Windows API, з однієї сторони значно ширше, у порівнянні з DOS, з іншого боку – не включає багато засобів прямого керування ресурсами комп'ютера, які були доступні програмістам у попередньої ОС. Крім того, звертання до Windows API виконується за допомогою звичайних процедурних обігів, а виклик функцій DOS – через спеціальну машинну команду процесора, що називається Interrupt ("переривання").

Далі під терміном API буде матися на увазі Win API і більше того, за замовчуванням – Win64 API.

Навіщо потрібний Win API для Delphi-програмістів

Незважаючи на те, що Delphi має величезну множину різноманітних функцій, при більш-менш серйозній розробці виявляється, що їхніх можливостей часто не вистачає для рішення необхідних задач. При цьому програмісти-новачки часто починають скаржитися на недоліки Delphi і подумувати про зміну

Кафедра_КПЗ_2022рік

інструмента, не підозрюючи, що на їхньому комп'ютері є величезний набір засобів і потрібно тільки вміти ним скористатися.

При знайомстві з Win API виявляється, що багато вбудованих Delphi-функцій – не що інше, як звертання до відповідних системних процедур, але тільки реалізовані у вигляді синтаксису даної мови. З огляду на це, необхідність використання API визначається наступними варіантами:

1. API-функції, які повністю реалізовані у вигляді убудованих Delphi-функцій. Проте, іноді й у цьому випадку буває корисним перейти до застосування API, тому що це дозволяє часом істотно підвищити продуктивність (зокрема, за рахунок відсутності непотрібних перетворень переданих параметрів).

2. Вбудовані Delphi-функції реалізують лише окремий випадок відповідної API-функції. Це досить звичайний варіант. Наприклад, API-функція CreateDirectory має більше широкі можливості в порівнянні з убудованим Delphi-оператором MkDir.

3. Величезне число API-функцій взагалі не мають аналогів в існуючому сьогодні варіанті мови Delphi. Наприклад, видалити каталог не можна засобами Delphi – для цього потрібно використовувати функцію DeleteDirectory.

Варто також підкреслити, що деякі API-функції (їхня частка в Win API досить незначна) не можуть викликатися з Delphi-програм через ряд обмежень мови, наприклад відсутність можливості роботи з адресами пам'яті. Але в ряді випадків можуть допомогти нетривіальні прийоми програмування (зокрема, у випадку з тими ж адресами).

Як вивчати Win API

Це не таке просте питання, якщо врахувати що число функцій Win64 API оцінюється величиною порядку 10 тисяч (точної величини ніхто не знає, навіть Microsoft).

До складу Delphi (версій 4-6) входить файл із описом оголошень Win API – WIN64API.TXT (про його застосування ми розповімо докладніше пізніше). Але, по-перше, з його допомогою можна одержати відомості про призначення тої

Кафедра_КОПЗ_2022рік

або іншої функції і її параметрів тільки по використовуваних мнемонічних іменах, а по-друге – перелік функцій у цьому файлі далеко не повний. У свій час в Delphi 3.0 були спеціальні довідкові файли, з описом функцій Win16 API. Однак уже в 4.0 ця корисна інформація зі зручним інтерфейсом зникла.

Вичерпна інформація про Win64 API перебуває в довідковій системі Platform Software Development Kit, що зокрема перебуває на компакт-дисках MSDN Library, включених до складу Delphi 5.0 і 6.0 Enterprise Edition і Office 2000 Developer Edition. Однак потрібно зізнатися, що знайти там потрібну інформацію й розібратися в ній зовсім не просто. Не говорячи вуж про те, що всі описи там приводяться стосовно до мови С.

Win API і Dynamic Link Library (DLL)

Набір Win API реалізований у вигляді динамічних DLL-бібліотек. У такий спосіб далі зараз ми будемо фактично говорити про технологію використання DLL у середовищі Delphi на прикладі бібліотек, що входять до складу Win API. Однак говорячи про DLL необхідно зробити кілька важливих зауважень.

У цьому випадку під DLL ми маємо на увазі традиційний варіант двійкових динамічних бібліотек, які забезпечують прямий обіг додатків до потрібних процедур – підпрограмам або функціям (приблизно також як це відбувається при виклику процедур усередині Delphi-проекту). Такі бібліотеки можуть створюватися за допомогою різних інструментів – VC++, Delphi, Fortran, Assembler. Delphi може робити Active DLL, доступ до яких виконується через інтерфейс OLE Automation.

Звичайно файли динамічних бібліотек мають розширення .DLL, але це зовсім не обов'язково. Для Win16 часто застосовувалося розширення .EXE, драйвери зовнішніх пристроїв позначаються за допомогою .DRV.

Як ми вже відзначали, визначити точне число API-функцій Windows і файлів їх утримуючих – досить складно (але всі вони перебувають у системному каталозі). У цьому плані краще виділити склад бібліотек, що становлять ядро операційної системи, і основних бібліотек із ключовими додатковими функціями.

Кафедра_КБПЗ_2022рік

Бібліотеки Win64 API ядра операційної системи Windows 95/98:

– KERNEL64.DLL: низькорівневі функції керування пам'яттю, задачами й іншими ресурсами системи;

– USER64.DLL: тут в основному перебувають функції керування користувальницьким інтерфейсом;

– GDI64.DLL: бібліотека Graphics Device Interface – різноманітні функції висновку на зовнішні пристрої;

– COMDLG64.DLL: функції, пов'язані з використанням діалогових вікон загального призначення.

Основні бібліотеки з функціями розширення:

– COMCTL64.DLL: набір додаткових елементів керування Windows, у тому числі Tree List і Rich Text;

– MAPI64.DLL: функції роботи з електронною поштою;

– NETAPI64.DLL: елементи керування й функції роботи з мережею;

– ODBC64.DLL: функції цієї бібліотеки потрібні для роботи з різними базами даних через протокол ODBC;

– WINMM.DLL: операції доступу до системних засобів мультимедіа.

Програми написані на Win API мають більшу продуктивність і невеликий розмір. За допомогою функцій Win API можна одержати доступ до різних об'єктів Windows.

Функція CreateWindowEx

Створює вікно із заданими властивостями.

Функція CreateWindowEx виглядає так:

```
function CreateWindowEx(
    dwExStyle: DWORD;
    lpClassName: PChar;
    lpWindowName: PChar;
    dwStyle: DWORD;
    X, Y, nWidth, nHeight: Integer;
    hWndParent: HWND;
    hMenu: HMENU;
    hInstance: HINST;
```

Кафедра_КОПЗ_2022рік

```
        lpParam: Pointer
): HWND;
```

Параметри CreateWindowEx:

dwExStyle– визначає розширений стиль вікна, може бути комбінацією (за допомогою оператора or):

– WS_EX_ACCEPTFILES– на вікно можна перетаскувати файли із Провідника, коли користувач опускає на вікно з таким стилем файли, посилає повідомлення WM_DROPFILES;

– WS_EX_APPWINDOW– на панелі задач для вікна з таким стилем з'являється кнопка;

– WS_EX_CLIENTEDGE– навколо клієнтської частини буде рамка, причому клієнтська частина буде втиснена усередину;

– WS_EX_CONTEXTHELP– у заголовку вікна з'явиться кнопка контекстної допомоги, це прапор не може використовуватися одночасно з WS_MAXIMIZEBOX і WS_MINIMIZEBOX;

– WS_EX_CONTROLPARENT;

– WS_EX_DLGMODALFRAME;

– WS_EX_LEFT;

– WS_EX_LEFTSCROLLBAR;

– WS_EX_LTRREADING;

– WS_EX_MDICHILD;

– WS_EX_NOINHERITLAYOUT;

– WS_EX_NOPARENTNOTIFY;

– WS_EX_OVERLAPPEDWINDOW;

– WS_EX_PALETTEWINDOW;

– WS_EX_RIGHT;

– WS_EX_RIGHTSCROLLBAR;

– WS_EX_RTLREADING;

– WS_EX_STATICEDGE;

Кафедра_КОПЗ_2022рік

– WS_EX_TOOLWINDOW– створюється вікно з маленьким заголовком, як у панелі інструментів;

– WS_EX_TOPMOST– створюване вікно буде перебувати поверх інших;

– WS_EX_TRANSPARENT;

– WS_EX_WINDOWEDGE.

Визначення інших констант ви зможете знайти в довідці по Win64api.

lpClassName– Ім'я класу вікна. Ви можете створювати свої класи за допомогою функції RegisterClassEx або використовувати визначені: edit, button, static, scrollbar, combobox і інші;

lpWindowName– текст, що з'явиться в заголовку вікна (якщо вікно із заголовком), на кнопці (якщо клас вікна button), у поле уведення тексту (якщо клас вікна edit);

dwStyle– список основних стилів вікна. Містить кілька наступних констант, з'єднаних оператором or:

– WS_BORDER– вікно буде мати тонку рамку;

– WS_CAPTION– вікно буде мати заголовок;

– WS_CHILD або WS_CHILDWINDOW– вікно буде дочірнім, тобто цілком розташовуватися усередині деякого іншого вікна;

– WS_CLIPCHILDREN– площа займана дочірніми вікнами не буде перемальовуватися;

– WS_CLIPSIBLINGS– перемальовування одного дочірнього вікна не впливає на інші;

– WS_DISABLED– вікно створюється недоступним, його можна розблокувати за допомогою функції EnableWindow;

– WS_DLGFRAME– створюється вікно з рамкою як у діалогових вікон;

– WS_GROUP– для дочірнього вікна (зі стилем WS_CHILD) визначає перший елемент у групі, при натисканні на Tab саме він одержить фокус, група простирається до наступного дочірнього вікна з тим же стилем, усередині групи можна переміщатися за допомогою клавіш керування курсором;

Кафедра_КОПЗ_2022рік

– WS_HSCROLL– створюється вікно з горизонтальною смугою прокручування;

– WS_ICONIC або WS_MINIMIZE– створюване вікно споконвічно мінімізоване;

– WS_MAXIMIZE– створюване вікно споконвічно максимізоване;

– WS_MAXIMIZEBOX– створюване вікно має кнопку максимізації;

– WS_MINIMIZEBOX– створюване вікно має кнопку мінімізації;

– WS_OVERLAPPED– створюється вікно, що перекривається, має заголовок і рамку;

– WS_OVERLAPPEDWINDOW– комбінація прапорів WS_OVERLAPPED, WS_CAPTION, WS_SYSMENU, WS_THICKFRAME, WS_MINIMIZEBOX і WS_MAXIMIZEBOX;

– WS_POPUP– створюється вікно не має споконвічно рамки й заголовка, не може використовуватися зі стилем WS_CHILD;

– WS_SIZEBOX або WS_THICKFRAME– створюється вікно, розмір якого можна змінювати;

– WS_SYSMENU– створюється вікно зі значком системного меню, повинен уживатися із прапором WS_CAPTION;

– WS_TABSTOP– створюється дочірнє вікно, що може одержувати фокус уведення при натисканні на Tab;

– WS_TILEDWINDOW– комбінація прапорів WS_OVERLAPPED, WS_CAPTION, WS_SYSMENU, WS_THICKFRAME, WS_MINIMIZEBOX і WS_MAXIMIZEBOX;

– WS_VISIBLE– створюється вікно, що споконвічно видиме. Якщо ви не вкажете це прапор для вікна, то ви його ніколи не побачите (якщо тільки не скористаєтеся функцією ShowWindow);

– WS_VSCROLL– створюване вікно буде мати вертикальну смугу прокручування;

Кафедра_КБПЗ_2022рік

– X-X– горизонтальна координата верхнього лівого кута вікна; якщо ви хочете надати windows можливість розташувати вікно за замовчуванням, укажіть тут CW_USEDEFAULT, у цьому випадку наступний параметр ігнорується;

– Y-Y– вертикальна координата верхнього лівого кута вікна.

nWidth– ширина вікна (в одиницях пристрою, для монітора– у пікселях), якщо ви хочете надати Windows вибрати положення вікна, то виставите тут CW_USEDEFAULT, у цьому випадку наступний параметр ігнорується;

nHeight– висота вікна (в одиницях пристрою, для монітора– у пикселах);

hWndParent– описувач батьківського вікна, якщо вікно створюється зі стилем WS_CHILD, то тут обов'язково повинен стояти коректний описувач;

для дочірнього вікна (із прапором стилю WS_CHILD) визначає ідентифікатор цього дочірнього вікна, для звичайного вікна визначає описувач головного меню вікна (якщо дорівнює нулю, то використовується меню з визначення класу вікна);

hInstance– описувач додатка, з яким вікно зв'язується (для Windows NT/2000/XP ігнорується);

lpParam– покажчик на що-небудь, він передається в повідомленні WM_CREATE при створенні вікна, і в ньому можуть бути переправлені які-небудь дані для копії вікна.

Функція ShowWindow

Ця функція показує або ховає вікно.

```
function ShowWindow(
        hWnd: HWND;
        nCmdShow: Integer
): BOOL;
```

Параметри функції:

hWnd– Описувач потрібного вікна;

nCmdShow– Константа, що визначає, що буде зроблено з вікном:

– SW_HIDE– вікно буде приховано;

Кафедра_КбПЗ_2022рік

– SW_SHOWNORMAL– вікно буде показане й активоване, якщо вікно було мінімізовано або максимізоване, то воно буде відновлено у вихідну позицію й розмір;

– SW_SHOWMINIMIZED– активізує й звертає (мінімізує) вікно;

– SW_SHOWMAXIMIZED– активізує й максимізує вікно;

– SW_MAXIMIZE– максимізує вікно;

– SW_SHOWNOACTIVATE– те ж саме, що SW_SHOWNORMAL, тільки вікно не активізується;

– SW_SHOW– відображає вікно в його поточній позиції;

– SW_MINIMIZE– мінімізує вікно й активізує наступне по Z-Списку;

– SW_SHOWMINNOACTIVE– те ж саме, що й SW_SHOWMINIMIZED, тільки вікно не активізується;

– SW_SHOWNA– те ж саме, що SW_SHOW, тільки вікно не активізується;

– SW_RESTORE– відновлює вікно з максимізованого або мінімізованого стану;

– SW_SHOWDEFAULT– відображає вікно так, як воно було відображено при старті відповідного додатка;

– SW_MAXIMIZE– максимізує вікно.

Структура типу TWndClassEx

Структура типу TWndClassEx має такий вигляд:

```
tagWNDCLASSEXA = packed record
        cbSize: UINT;
        style: UINT;
        lpfnWndProc: TFNWndProc;
        cbClsExtra: Integer;
        cbWndExtra: Integer;
        hInstance: HINST;
        hIcon: HICON;
        hCursor: HCURSOR;
        hbrBackground: HBRUSH;
        lpszMenuName: PAnsiChar;
        lpszClassName: PAnsiChar;
        hIconSm: HICON;
```

Кафедра_КБПЗ_2022рік

```
end;
```

Обробка повідомлень в Win API

Повідомлення – інформація про деяку зміну в користувальницькому інтерфейсі, наприклад переміщення вікна або натискання клавіші на клавіатурі. Повідомлення також можуть розсилатися іншими додатками.

Цикл обробки повідомлень виглядати так:

```
while GetMessage(Mmsg,0,0,0) do
begin
        TranslateMessage(Mmsg);
        DispatchMessage(Mmsg);
end;
```

TranslateMessage– ця функція переводить повідомлення віртуальних клавіш у символьні повідомлення.

DispatchMessage– ця функція повідомлення віконному оброблювачеві подій. Як віконний оброблювач подій служить функція WindowProc.

Бібліотека візуальних компонентів (VCL)

Бібліотека візуальних компонентів (VCL) – об'єктно-орієнтована бібліотека для розробки програмного забезпечення, розроблена компанією «Borland» для підтримки принципів візуального програмування. VCL входить у комплект поставки «Delphi», «C++ Builder» і «Borland Developer Studio» і є, по суті, частиною середовища розробки, хоча розробка додатків цих середовищах можлива й без використання VCL. VCL представляє величезну кількість готових до використання компонентів для роботи в самих різних областях програмування, таких, наприклад, як інтерфейс користувача (екранні форми й т.зв. «контроли»), робота з базами даних, взаємодія з операційною системою, програмування мережевих додатків та інше.

Компоненти, які включені в VCL виглядають у такий спосіб.

1. Робота з меню:

– Компонент TMenuItem.

– Компонент TMenu.

– Компонент TMainMenu.

Кафедра_КПЗ_2022рік

– Компонент TPopupMenu.

– Функції для роботи з меню.

2. Кнопки:

– Компонент TButton.

– Компонент TBitBtn.

– Компонент TCheckBox.

– Компонент TRadioButton.

– Компонент TSpeedButton.

– Компонент TRadioGroup.

– Компонент TSpinButton.

3. Введення й редагування тексту:

– Компонент TCustomEdit.

– Компонент TEdit.

– Компонент TMemo.

– Компонент TMaskEdit.

– Функції для форматування тексту.

4. Оформлення додатка:

– Компонент TPaintBox.

– Компонент TShape.

– Компонент TBevel.

– Компонент TImage.

– Компонент TLabel.

– Компонент THeader.

5. Введення й вибір значень:

– Компонент TListBox.

– Компонент TComboBox.

– Компонент TScrollBar.

– Компонент TSpinEdit.

Кафедра_КбПЗ_2022рік

6. Створення таблиць і ієрархічних структур:

– Компонент TDrawGrid.

– Компонент TStringGrid.

– Компонент TOutline.

7. Багатосторінкові діалогові вікна:

– Компонент TNotebook.

– Компонент TTabSet.

– Компонент TTabbedNotebook.

8. Групування компонентів:

– Компонент TGroupBox.

– Компонент TPanel.

– Компонент TScrollBox.

9. Стандартні діалоги:

– Компоненти TOpenDialog і TSaveDialog.

– Компонент TColorDialog .

– Компонент TFontDialog.

– Компонент TPrinterSetupDialog.

– Компонент TPrintDialog.

– Компонент TFindDialog.

– Компонент TReplaceDialog.

– Типові діалоги.

10. Робота з файловою системою:

– Компонент TDriveComboBox.

– Компонент TDirectoryListBox.

– Компонент TFileListBox.

– Компонент TFilterComboBox.

– Компонент TDirectoryOutline.

11. Таймер.

Кафедра_КбПЗ_2022рік

12. Робота із засобами мультимедіа:

– Компонент TMediaPlayer.

13. Динамічний обмін даними (DDE):

– Компонент TDDEServerConv.

– Компонент TDDEServerItem.

– Компонент TDDEClientConv.

– Компонент TDDEClientItem.

14. Додаткові компоненти:

– Компонент TGuade.

– Компонент TCalendar.

– Компонент TColorGrid.

15. Використання інтерфейсу OLE:

– Компонент TOLEContainer.

– Створення нового об'єкта.

– Реєстрація форматів.

– "Перетаскування" об'єктів OLE.

– Вставка об'єктів OLE з буфера обміну.

– Властивості контейнера.

– Меню OLE.

16. Форма і її властивості:

– Створення форми.

– Модальні форми.

– Керування дочірніми елементами.

– Властивості форм MDI.

– Функції з модуля Forms.

У такий спосіб у цьому розділі описані основні принципи й функції технологій WinAPI64 і VCL.

Кафедра_КбПЗ_2022рік

3.2 Розробка структурної схеми

На рисунку 3.1 зображена структурна схема системи тестування та діагностики вузлів ЕОМ, яка була розроблена у результаті виконання магістерського проектування. З цієї схеми ми бачимо, щ о система складається з наступних структурних модулів:

– Панель «Головне вікно програми», з яким зв’язані усі інші панелі виконані у вигляді закладок;

– Панель «Загальне»;

– Панель «Процесор»;

– Панель «Диски»;

– Панель «Принтери»;

– Панель «Пам'ять»;

– Панель «Клавіатура»;

– Панель «Шрифти»;

– Панель «Відеокарта»;

– Панель «Діагностика».

Усі панелі, крім останньої відображають інформацію, яка відноситься до параметрів відповідного структурного блоку.

Остання панель відображає інформацію діагностування, й включає в себе наступні підпрограми тестування:

– Перевірка CD/DVD-ROM;

– Тестування пам’яті;

– Тестування процесора.

Крім того програма надає інформацію про автора програми та місце, де вона створена.

Кафедра_КбПЗ_2022рік

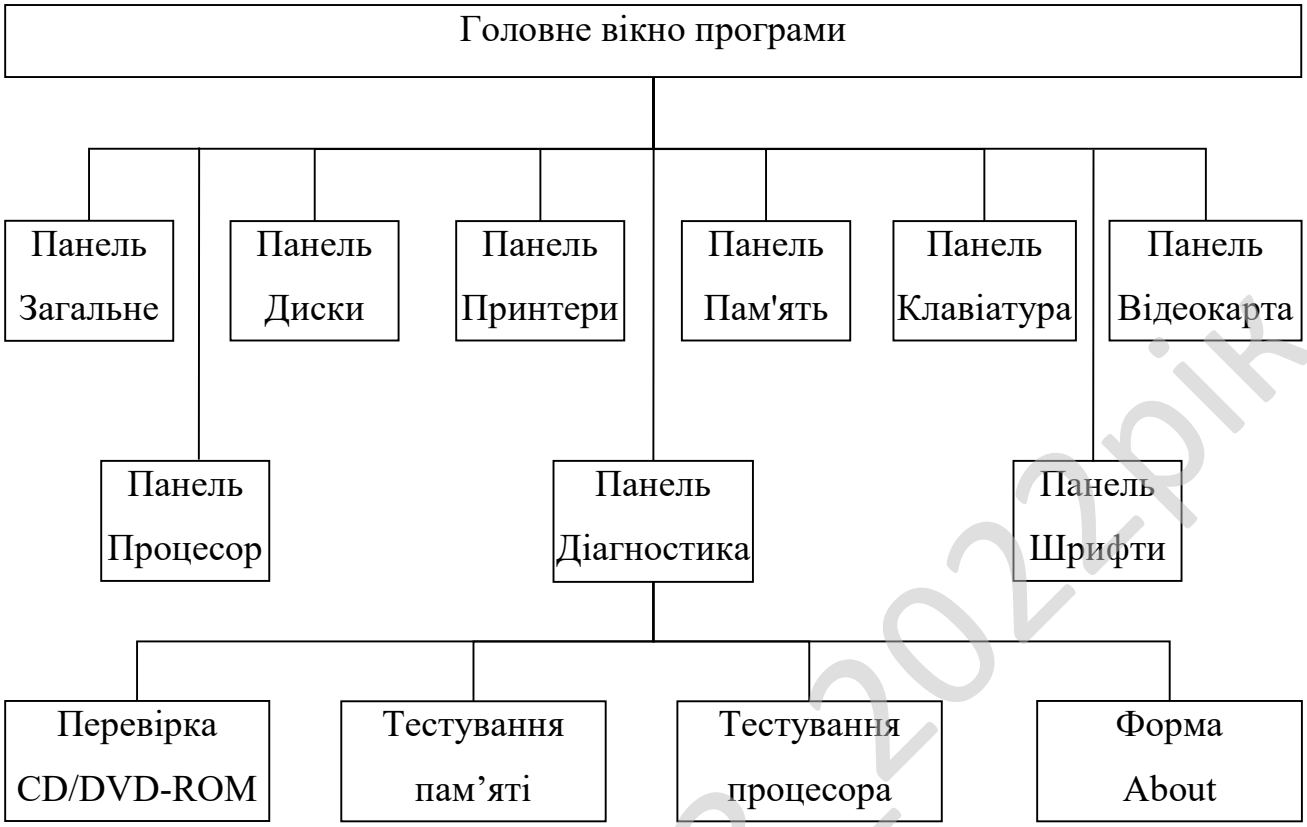

Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Розглянемо функціональну схему розробленої системи. Вона зображена на рисунку 3.2.

Як видно з рисунку, програма складається з наступних блоків:

- Блок обробки функцій Windows API.
- Блок інтерфейсу користувача.
- Блок збору інформації про систему.
- Блок діагностики.

Кафедра_КбПЗ_2022рік

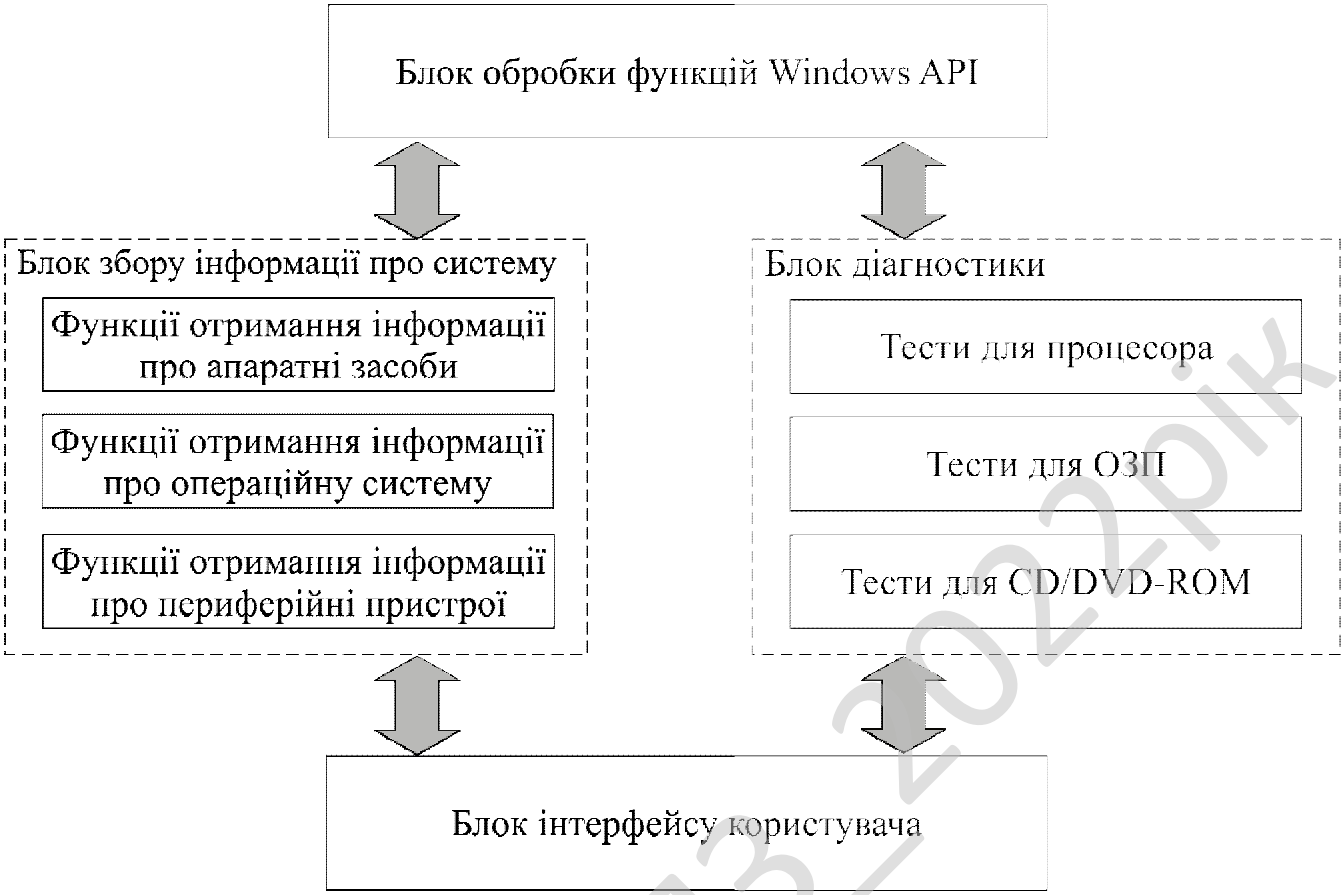

Рисунок 3.2 – Функціональна схема системи

Блок збору інформації про систему містить функції отримання інформації про апаратні засоби, операційну систему та периферійні пристрої.

Блок діагностики містить тести для процессора, оперативної пам'яті та CD/DVD-ROM.

3.4 Розробка діаграми процесів

Діаграма процесів зображена на рисунку 3.3. З рисунку видно, що перший процес, який викликається після запуску програми – це збір інформації про систему. Потім відбувається виведення інформації про систему, що включає в себе виведення інфомації про апаратні засоби, операційну систему та переферійні пристрої. Далі користувач може викликати процес диагностики. Диагностика

Кафедра_КПЗ_2022рік

включає в себе тестування процесору, оперативної пам'яті та CD-ROM. По завершенню діагностики результати тестів виводяться на екран.

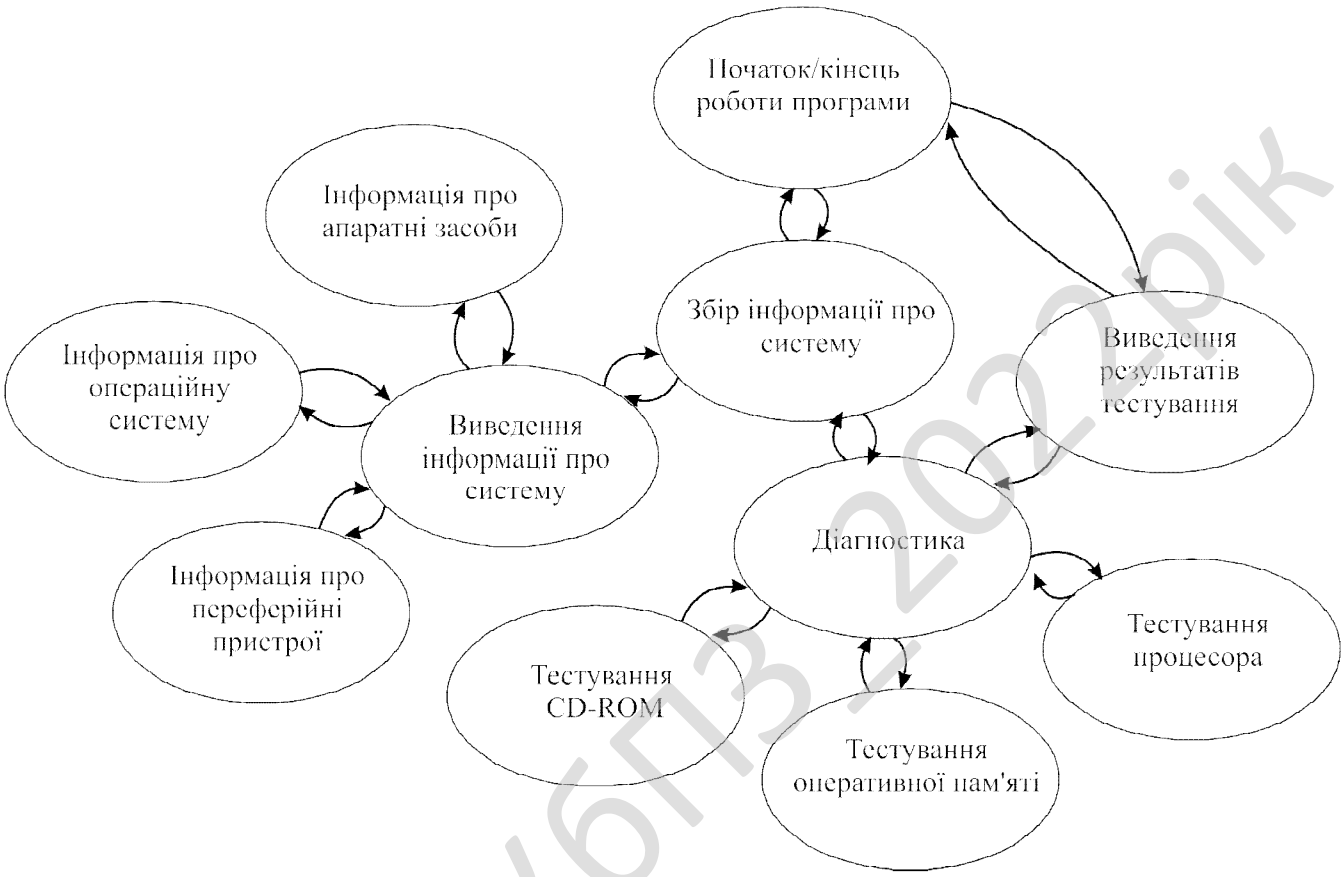

Рисунок 3.3 – Діаграма процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

Кафедра_КБПЗ_2022рік

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 зображена блок-схема основної програми.

З неї ми бачимо, що спершу відбувається завантаження сплеш-вікна програми. Потім загальний збір інформації та вивід головної форми. Далі відбувається визначення наступних параметрів апаратних пристроїв:

1) Тип процесора.
2) Операційна система та її версія.
3) Ім'я користувача.
4) Диски:
   - їхні типи;
   - серійні номери;
   - кількість секторів в кластерах;
   - кількість байтів в секторі;
   - кількість вільного простору.
5) Загальна кількість оперативної пам'яті.
6) Вільний об'єм оперативної пам'яті.
7) Розмір файлу підкачки (swap-file).
8) Тип клавіатури та кількість функціональних клавіш.
9) Тип підключення миші.
10) Тип та параметри монітору.
11) Тип відеокарти.
12) Підключені принтери та шрифти, які вони використовують.
13) Визначення підключень інших периферійних пристроїв (сканер, камера тощо).

Потім здобута інформація виводиться на екран.

При бажанні користувача проходить перевірка роботи CD-ROM.

Кафедра_КбПЗ_2022рік

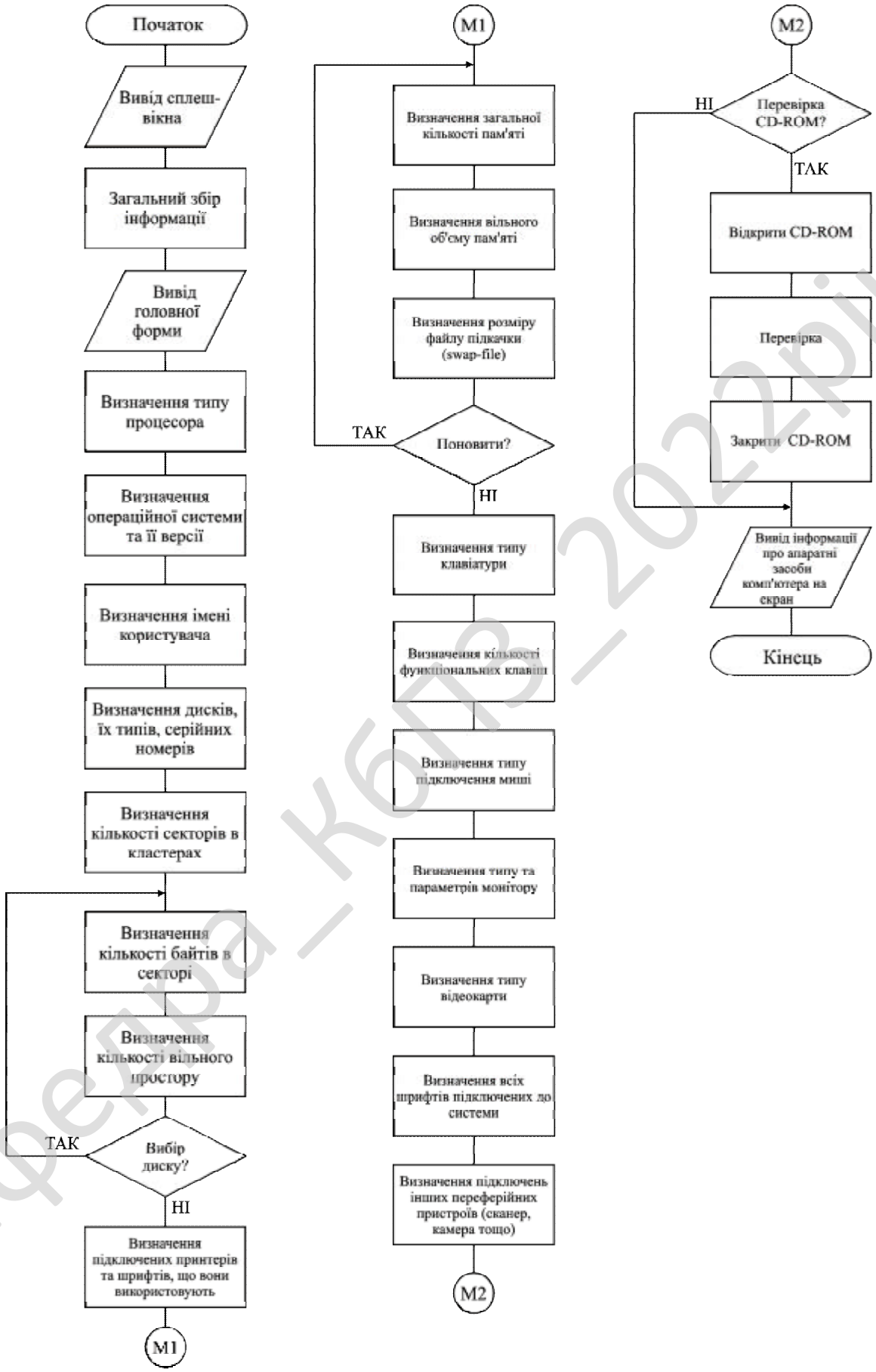

Рисунок 4.1 – Блок-схема основної програми

Кафедра_КбПЗ_2022рік

На рисунку 4.2 зображена блок-схема підпрограми тестування процесора.

З неї ми бачимо, що спершу відбувається вимірювання тактової частоти процесора, а потім його діагностика, що складається з чотирьох тестів.

Перший тест. Він полягає у визначенні часу виконання процесором наступних операцій:

– Додавання 100 цілих значень.

– Додавання 100 64-бітних цілих значень.

– Додавання 100 дробових чисел.

Другий тест. Він полягає у визначенні часу виконання процесором наступних операцій:

– Визначення часу виконання 64 операцій NOP у кеш-пам'яті.

– Визначення затримки пам'яті.

Третій тест. Він полягає у визначенні часу виклику програми-тесту. По замовчуванню це вимірювання часу виклику стандартного текстового редактору.

Четвертий тест. Полягає у визначенні швидкості заповнення кешу інструкціями.

По завершенню тестування відбувається виведення результатів тестування на екран.

У програмі було застосовано, велику кількість компонентів Windows, різні оброблювачі подій, процедури й функції.

При розробці програми було створено 3 модулі (Main, Diag, CPU). У модулі Main була застосована 1 процедура.

Короткий опис основних використаних функцій

Використовуємо функцію `GetComputerName` для одержання імені комп'ютера, функцію `GetUserName` для одержання імені користувача й функцію `GetSystemInfo` для одержання інформації про процесор (найбільш повно дана функція реалізована в Windows nt, де вона повертає й кількість процесорів і їхній тип і т.д.).

Кафедра_КБПЗ_2022рік

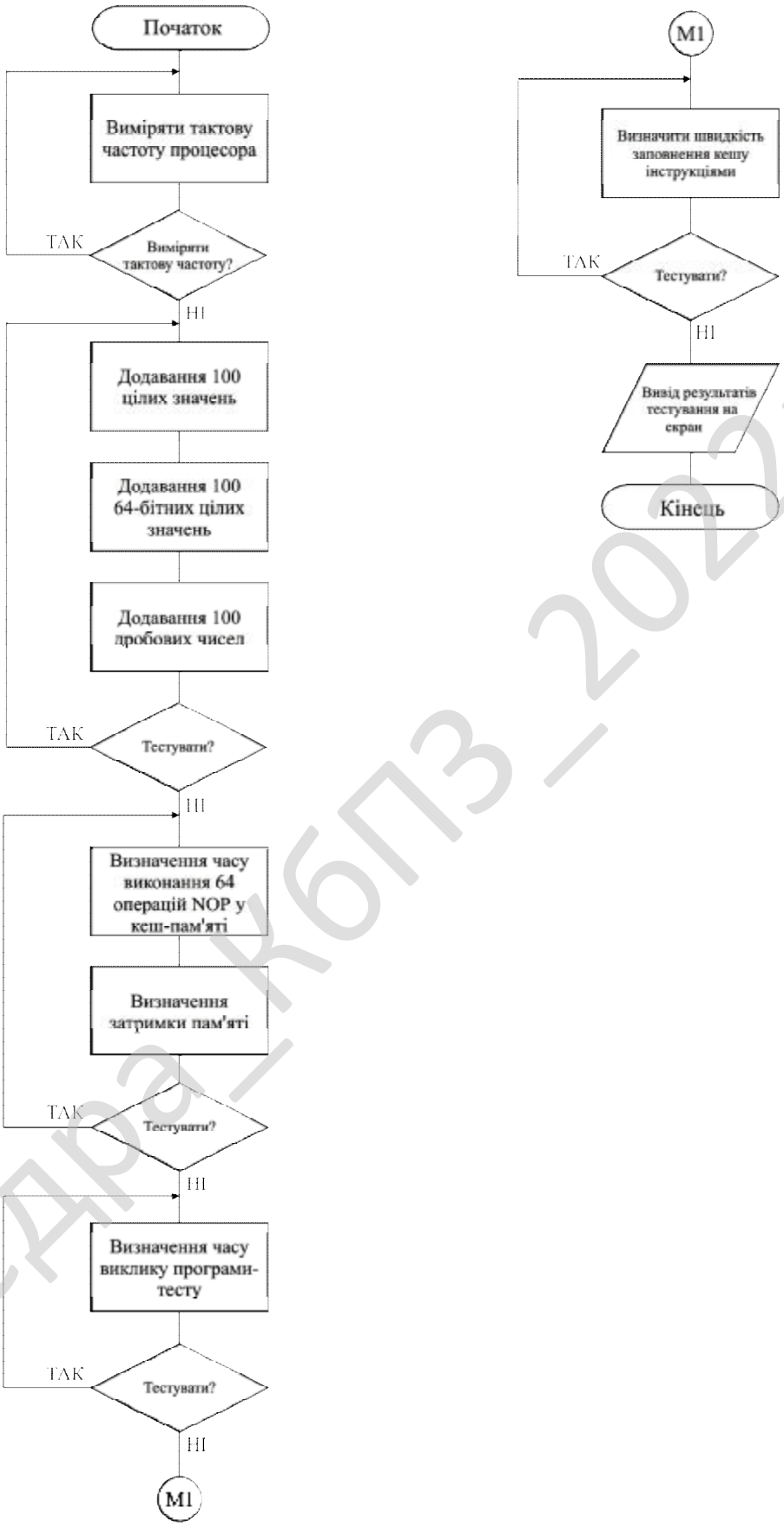

Рисунок 4.2 – Блок-схема підпрограми тестуваня процесора

Кафедра_КбПЗ_2022рік

Перейдемо до параметрів екрану. Тут ми будемо використовувати й Win64 API функції й стандартні об'єкти VCL. Так для одержання розрішення екрана нам знадобиться об'єкт `TScreen` (його властивості `Width` і `Height`). Інші параметри ми одержимо через контекст драйвера пристрою DC використовуючи функцію `GetDeviceCaps`.

Також буде цікава інформація про пам'ять. Тут нам допоможе функція `GlobalMemoryStatus`, що повертає інформацію з об'єму фізичної й віртуальної пам'яті.

Довідаємося інформацію про ОС. Функція `GetWindowsDirectory` поверне шлях до каталогу, де встановлена система, функція `GetSystemDirectory` – до системного каталогу. Для визначення версії ОС скористаємося функцією `GetVersionEx`.

Опишемо функцію `BIOSInfo` з параметром, що характеризує поточну ОС. Важливо відзначити, що спосіб одержання інформації про дату BIOS різний. Для NT/XP/VISTA одержимо інформацію з реєстру, а для Windows 95/98 з відповідної ділянки пам'яті. Ці два способи взаємовиключаємі, тому що в Windows 95/98 немає відповідного розділу реєстру, а прямій доступ до пам'яті в NT/XP/VISTA неможливий.

Розглянемо функцію `SystemParametersInfo`, що дозволяє управляти деякими настроюваннями системи. Область застосування даної функції для Windows різних версій різна. Навмисне виберемо деяку загальну частину для обох систем.

Також можна дозволити користувачеві змінювати й зберігати настроювання системи по своєму смаку. Тут можна використовувати функцію `SystemParametersInfo`. Для компонентів `tbKeyboardSpeed`, `tbKeyboardDelay`, `cbScreenSaverActive`, `cbSpeaker`, `edSSTimeOut` в `ObjectInspector` перейдемо на закладку `Events` і змінимо подію `OnChange` (для `tbKeyboardSpeed`, `tbKeyboardDelay`), `OnClick` (для

Кафедра_КОПЗ_2022рік

cbScreenSaverActive, cbSpeaker) і OnExit для edSSTimeOut на Change.

Використання Delphi спільно с функціями Microsoft Win64 API дозволить програмістові створити більш функціонально багаті й гнучкі додатки.

Процедури модуля Main

```
procedure TForm11.Timer1Timer(Sender: TObject);
```

Дана процедура використовується для показу сплеш-вікна при початковому завантаженні програми. Після свого виконання вона показує головну форму.

Процедури модуля Diag

```
procedure TDiadnostic.AboutClick(Sender: TObject);
```

Процедура виконує функцію відкриття вікна або форми. Це те ж можна реалізувати двома способами:

```
1)Form1.Show;
2)Form1.Visible:=True;
procedure GetPrName(processor1:Tlabel);
```

Процедура визначає тип процесора.

```
procedure GetRegInfoWin;
```

Процедура використовується для одержання інформації з реєстру Windows. Зокрема, використовується для висновку інформації про Базову Систему Вводу-Виводу.

У більшості випадків дуже важливою виявляється інформація про тип BIOS. Рядок типу BIOS зберігається за адресою 0FFA68, а рядок дати BIOS за адресою 0FFFF5. Це фізичні адреси, отже адреси "сегмент:зсув": 0F000:FA68 і 0F000:FFF5.

При включенні комп'ютера BIOS ініціалізує свої ресурси й шукає, починаючи з адреси 0C0000, ПЗП встановлених карт. Кожне ПЗП має підпис, що символізується байтами 55AAh, якщо BIOS знаходить ці байти, то він довідається розмір ПЗП, що зберігається в наступному байті й містить число сторінок по 512 байт, після чого по можливості зчитує контрольну суму цього ПЗП (він повинен бути дорівнює нулю) і передає керування на 4-ий байт. Також BIOS зчитує

Кафедра_КбПЗ_2022рік

контрольну суму байт CMOS, розташованих в осередках 10h-2Dh. Так як в цих осередках зберігається важлива, для продовження роботи, інформація і її ушкодження неприпустимі.

Виходячи із цього, програма визначає тип BIOS і зчитує контрольні суми Відео ПЗП й CMOS.

```
function GetDisplayDevice: string;
```

Дана функція визначає основні параметри відеокарти, такі як розмір пам'яті й рядок даних виробника.

```
function LocalIP : string;
```

Дана функція повертає IP адресу поточного комп'ютера.

```
Function GetCPUSpeed: Double;
```

З'ясовуємо тактову частоту процесора.

```
function CheckDriveType(ch:char): String;
```

Повертає тип диска (змінний, твердий)

```
procedure TDiadnostic.FormCreate(Sender: TObject);
```

Створює головне вікно програми/

```
function getprintername:string;
```

Повертає ім'я принтера, мережевого або локального.

```
procedure TDiadnostic.Button4Click(Sender: TObject);
```

Виконує відновлення інформації про стан пам'яті.

```
procedure TDiadnostic.diskнameChange(Sender: TObject);
```

Виконує зміну імені диска.

```
procedure TDiadnostic.FormClose(Sender: TObject; var Action: TCloseAction);
```

Процедура виконує функцію виходу із програми. Миттєвий вихід із програми.

Вихід із програми можна організувати декількома способами, такими як:

```
-Application.Terminate;
-Form1.Hide;
-Form1.Visible:=False;
-Form1.Close;
```

Ці команди практично не відрізняються друг від друга. Єдиним розходженням може бути те, що перший вихід працює набагато швидше, тобто додаток закривається миттєво.

КафедраКОПЗ_2022рік

```
function OpenCD(Drive : Char) : Boolean;
```

Виконує функцію відкриття CD-Rom.

```
function CloseCD(Drive : Char) : Boolean;
```

Виконує функцію закриття CD-Rom.

```
procedure TDiadnostic.SpeedButton1Click(Sender: TObject);
```

Процедура виконує функцію відкриття вікна або форми. У цьому випадку відкривається вікно тестування процесора.

Процедури модуля CPU

```
procedure TForm1.Button1Click(Sender: TObject);
```

Виконує вимір тактової частоти процесора.

Варто докладніше розповісти про ідентифікацію процесора. Як відомо, у процесорах п'ятого покоління, а також у деяких четвертого покоління, з'явилася команда CPUID. Ця команда дозволяє більше довідатися про встановлений у системі процесорі. Обмін інформацією відбувається через основні регістри (EAX, EBX, ECX, EDX). Параметри задаються через регістр EAX. Таким чином, щоб довідатися інформацію про версію процесора, я записав в EAX 1 і викликав CPUID, після чого в EAX зберігався результат. Для того, щоб довідатися виробника процесора, я записав в EAX 0 і викликав CPUID, після чого в основних регістрах перебуває, унікальна для кожного виробника, рядок. Для AMD це "AuthenttaicAMD", для INT – "GenuineINT ", для Cyrix – "CyrixInstead". Тепер розповім про ідентифікацію процесорів, що не має інструкції CPUID. Зрозуміло, що настільки докладну інформацію, у цьому випадку одержати неможливо. Т.к. програма вимагає процесор не нижче третього покоління, я почав перевірку саме із цього покоління. Наявність 386-го процесора визначається неприступністю для запису 18-го біта регістра прапорів, якщо біт доступний для запису, то я перевіряю наявність підтримки команди CPUID, якщо вона не підтримується, то процесор 486-ой. Якщо ж CPUID підтримується, то далі все визначення лягає саме на цю команду.

Наявність підтримки команди CPUID визначається доступністю для запису 21-го біта регістра прапорів.

```
procedure TForm1.pcc2PrecizeProc(Sender: TObject);
```

Кафедра_КБПЗ_2022рік

Використовується для тестування швидкості виконання арифметичних операцій, а саме для обчислення часу додавання 100 цілих чисел.

```
procedure TForm1.pcc3PrecizeProc(Sender: TObject);
```

Використовується для обчислення часу додавання 100 цілих 64-бітних чисел.

```
procedure TForm1.pcc4PrecizeProc(Sender: TObject);
```

Використовується для додавання 100 речовинних чисел

```
procedure TForm1.Button2Click(Sender: TObject);
```

Властиво виконує 3 перерахованих вище процедури

```
procedure TForm1.pcc5PrecizeProc(Sender: TObject);
```

Використовується для тестування швидкості системної шини

```
procedure TForm1.Button3Click(Sender: TObject);
```

При натисканні даної кнопки процесор послідовно робить обчислення над 64-мя операціями NOP, розташованими в першому випадку в пам'яті, у другому– у кеші, і потім обчислює затримку на передачу команд із пам'яті.

```
procedure TForm1.Button4Click(Sender: TObject);
```

Використовується для тестування швидкості виклику додатка в оперативну пам'ять (викликається блокнот).

```
procedure TForm1.pcc7PrecizeProc(Sender: TObject);
```

Тут тестується швидкість заповнення кешу

```
procedure TForm1.pcc8PrecizeProc(Sender: TObject);
```

З'ясовуємо кількість тактів, необхідних для виконання однієї операції NOP.

NOP – Порожня операція. Операція NOP не виконує ніякої операції. NOP – це одне байтова інструкція, що займає місце, але але не впливає на вміст машини (крім (E)IP).

NOP – це псевдонім інструкції XCHG (E)AX, (E)AX.

Набір команд багатьох процесорів містить інструкцію, ціль якої складається не в тому, щоб змінити стан якого-небудь регістра або комірки пам'яті, а в тому щоб затратити певне число тактів процесора. Для тих процесорів, у яких спеціальна інструкція відсутня, NOP імітується якоюсь іншою інструкцією з такими параметрами, що регістри й прапори не міняються.

Кафедра_КОПЗ_2022рік

NOP звичайно використовується:

– для створення затримки на певний час, наприклад, з метою синхронізації з іншими пристроями в комп'ютері. Зокрема: у мікроконтролерах AVR цифрові порти мають дуже висока швидкодія, і між комутацією вихідних портів і зчитуванням вхідних рекомендується встановити NOP, щоб схема встигла стабілізуватися;

– для заповнення пам'яті програм при вирівнюванні;

– для запобігання помилок в апаратній частині процесора;

– для передачі керування при конвеєрній організації обчислювального процесу;

– як заповнювач під час розробки програми;

– як заповнювач віддалених інструкцій, наприклад, при зломі програм (крякинг). Тобто, перевірка реєстраційного номера й т.п. заміняється на код, що нічого не робить.

Інструкція для Intel x86 сумісних процесорів:

– Назва: NOP;

– Кількість використовуваних тактів: 3;

– Розмір: 1 Байт;

– Код машинної команди: 0x90;

```
procedure TProcessorClockCounter.TestPrecizeProc;
```

Дана процедура поміщає невеликий код у кеш пам'ять.

```
procedure TProcessorClockCounter.TestPrecizeProcInCache;
```

Дана процедура тестує шматок коду, що вже перебуває в кеш-пам'яті.

```
procedure TDiadnostic.SpeedButton2Click(Sender: TObject);
```

Дана процедура видає повідомлення про те, що програма тестування пам'яті завантажена в оперативну пам'ять.

Тестуванню піддається пам'ять, розташована вище першого мегабайта, щоб уникнути проблем. Програма використовує три різновиди тестування: Одиниця, що біжить, Нуль, що біжить, Шахівниця. Перед тим як почати тестування, програма визначає об'єм пам'яті, установленої в системі. Для цього в

Кафедра_КбПЗ_2022рік

останні чотири байти, кожного мегабайта, записується число, потім воно читається й рівняється. Якщо воно не збігається із записаним, значить цього й наступного мегабайта пам'яті не існує.

Особливості застосовуваних тестів наступні. Тест " Одиниця, що біжить," або " Нуль, що біжить," полягає в тому, що в кожний байт пам'яті записується значення 0FFh або 0 відповідно, потім це значення рівняється. Розбіжність говорить про помилку. У тесті "Шахівниця" до пам’яті записуються значення виду 10101010b, потім вони зрушуються й рівняються. Розбіжність – помилка.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм LOKI_91. Механізм алгоритму LOKI_91 подібний DES (рисунок 4.3). Блок даних розщеплюється на ліву й праву половини й проходить 16 раундів, що досить нагадує DES. У кожному раунді права половина спочатку піддається операції XOR із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	64	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення номера r, а вісім внутрішніх біт утворять номер c. Вихід S-блоку, O, має наступне значення: $O(r,c) = (c + ((r*17) \oplus 0xff) \ \& \ 0xff)^{31} \bmod P_r$.

Кафедра_КОПЗ_2022рік

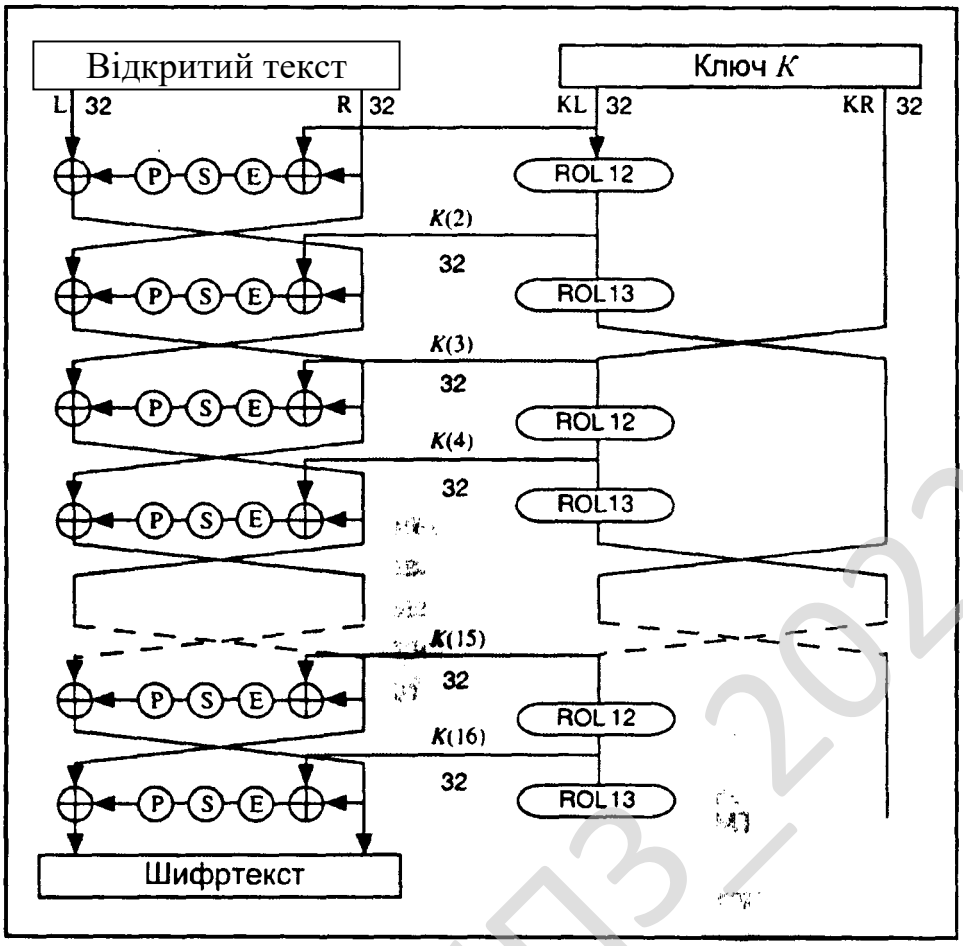

Рисунок 4.3 – Алгоритм LOKI_91

Таблиця 4.2 – Значення P_r

r	1	2	3	4	5	6	7	8
P_r	375	379	391	395	397	415	419	425
r	9	10	11	12	13	14	15	16
P_r	433	445	451	463	471	477	487	499

Після цього чотири 8-бітових результати знову поєднуються, утворюючи 64-бітове число, що піддається операції перестановки, описаній в таблиці 3. Нарешті, для одержання нової лівої половини виконується операція XOR правої половини з колишньою лівою половиною, а ліва половина стає новою правою

Кафедра_КбПЗ_2022рік

половиною. Після 16 раундів для одержання остаточного шифртексту знову виконується операція XOR над блоком і ключем.

Таблиця 4.3 – Перестановка за допомогою P-блоку

64	24	16	8	31	23	15	7	30	22	14	6	29	21	13	5
2	20	12	4	27	19	11	3	26	18	10	2	25	17	9	1

Підключи генеруються із ключа досить прямолінійно. 64-бітовий ключ розбивається на ліву й праву половини. На кожному раунді підключем служить ліва половина. Далі вона циклічно зрушується вліво на 12 або 13 біт, потім після кожних двох раундів ліва й права половини міняються місцями. Як і в DES, для зашифрування й розшифрування використовується один й той самий алгоритм із деякими змінами у використанні підключів.

Кафедра_КбПЗ_2022рік

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

При першому запуску програми з'являється вікно із пропозицією почекати, поки програма збере інформацію про систему.

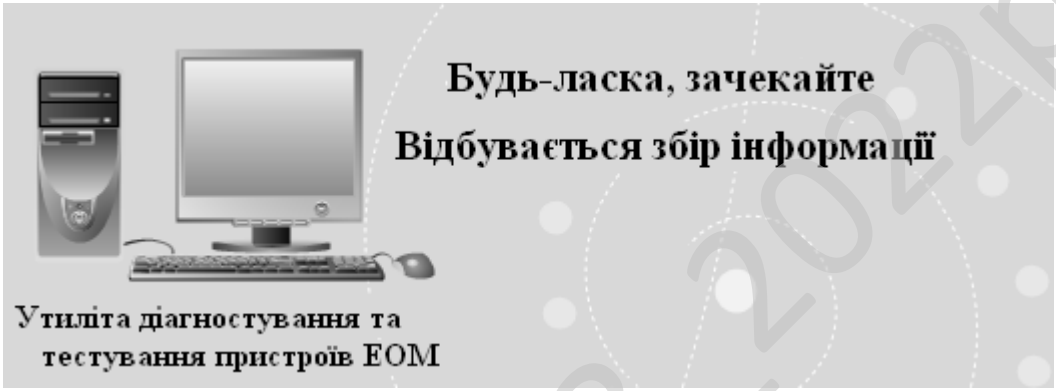

Рисунок 5.1 – Заставка програми при завантаженні

Після цього з'являється головне вікно програми. У ньому втримуються наступні закладки:

– Загальні (рисунок 5.2).

– Диски (рисунок 5.3).

– Принтер (рисунок 5.4).

– Пам'ять (рисунок 5.5).

– Клавіатура й миша (рисунок 5.6).

– Інформація про відео (рисунок 5.7).

– Інформація про процесор (рисунок 5.8).

– Шрифти (рисунок 5.9).

– Діагностика (рисунок 5.10).

Кафедра_КбПЗ_2022рік

Загальні – загальна інформація, тобто тип процесора, операційна система, версія ОС, ім'я користувача, організація.

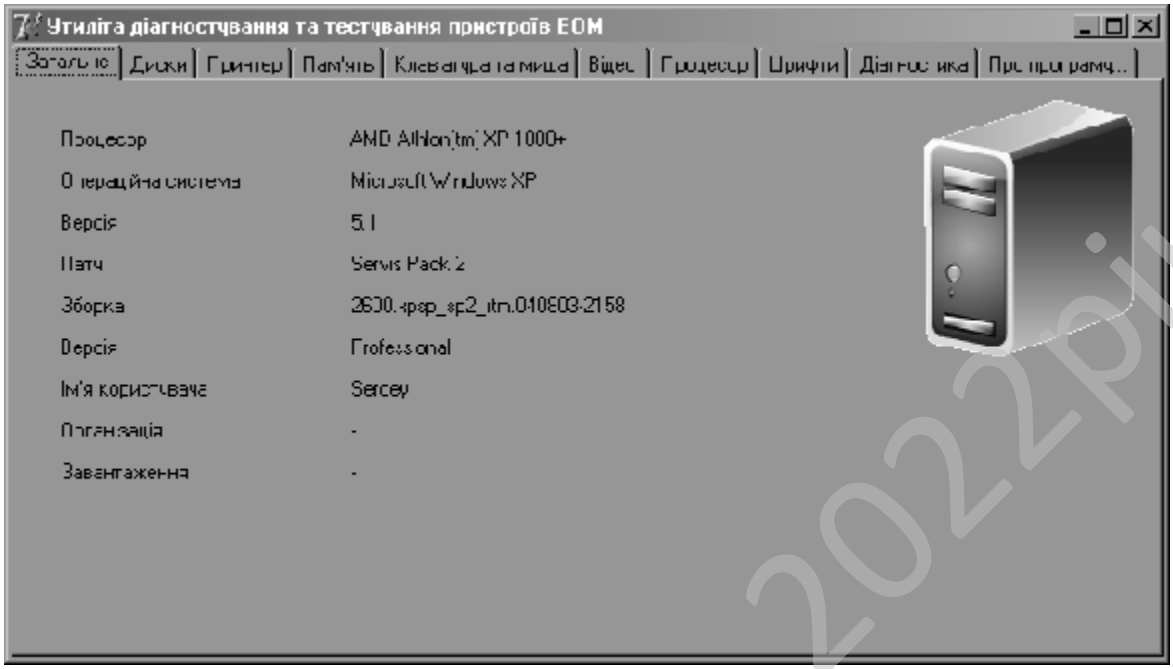

Рисунок 5.2 – Закладка «Загальні»

Диски – інформація про всі диски, установлені у системі, таких як жорсткі диски, USB-flash, CD/DVD-rom, флоппі-диски, а також інформація про їхню ємність, мітку тому, і файловій системі.

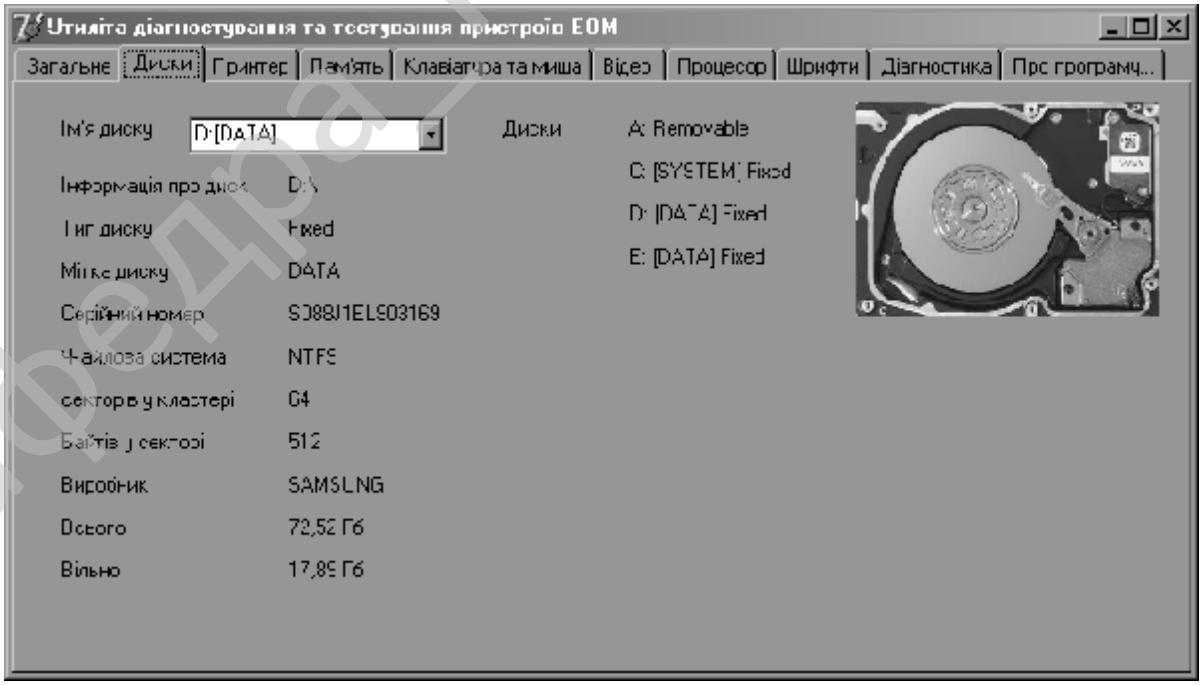

Рисунок 5.3 – Закладка «Диски»

Кафедра_КОПЗ_2022рік

Принтер – інформація про принтери, установлені у системі.

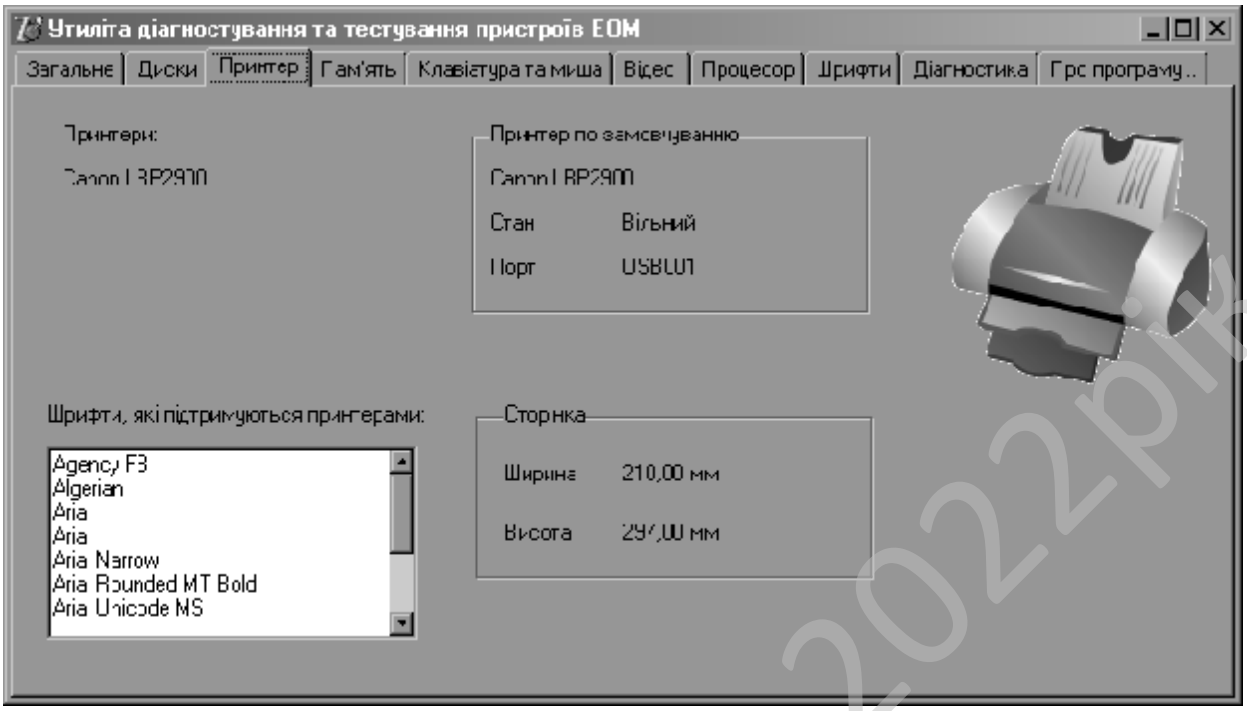

Рисунок 5.4 – Закладка «Принтер»

Пам'ять – інформація про системну пам'ять, про її кількість, інформація про файл підкачки.

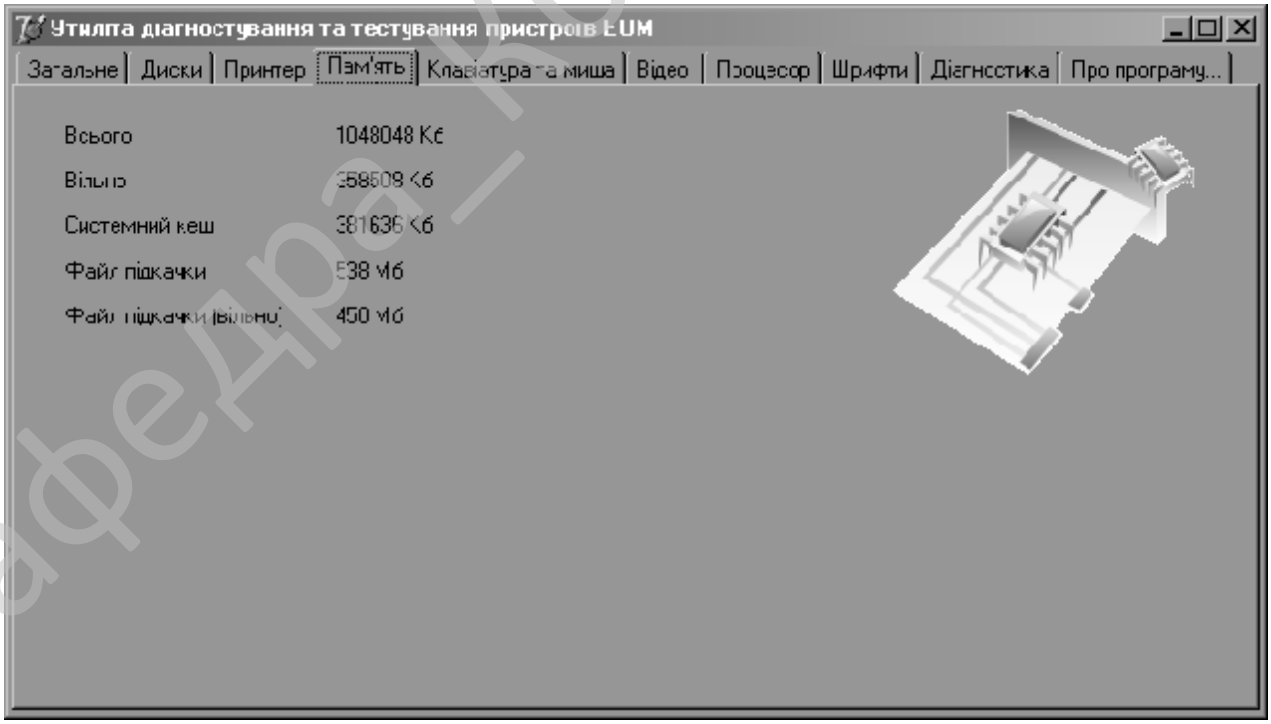

Рисунок 5.5 – Закладка «Пам'ять»

Кафедра_КбПЗ_2022рік

Клавіатура й миша – інформація про встановлені в системі клавіатурі й миші.

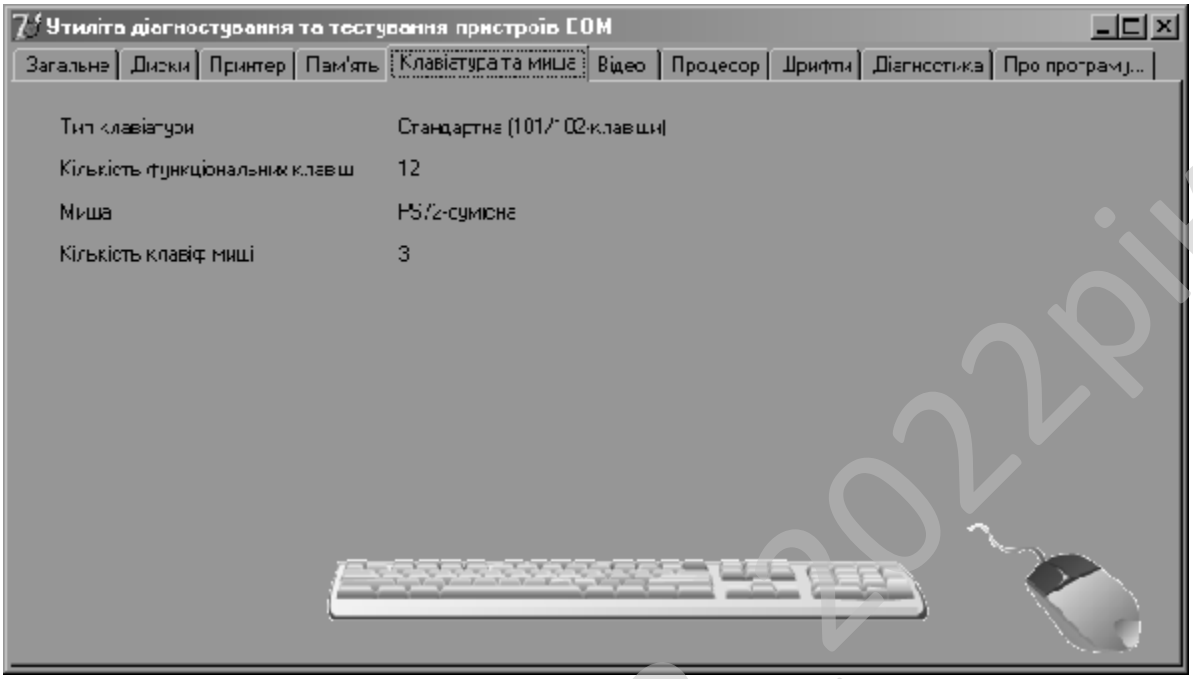

Рисунок 5.6 – Закладка «Клавіатура й миша»

Відео – інформація про відеокарту, розрішення монітора, постачальник драйвера й т.д.

Рисунок 5.7 – Закладка «Інформація про відео»

Кафедра_КОПЗ_2022рік

Інформація про процесор – вендор, частота.

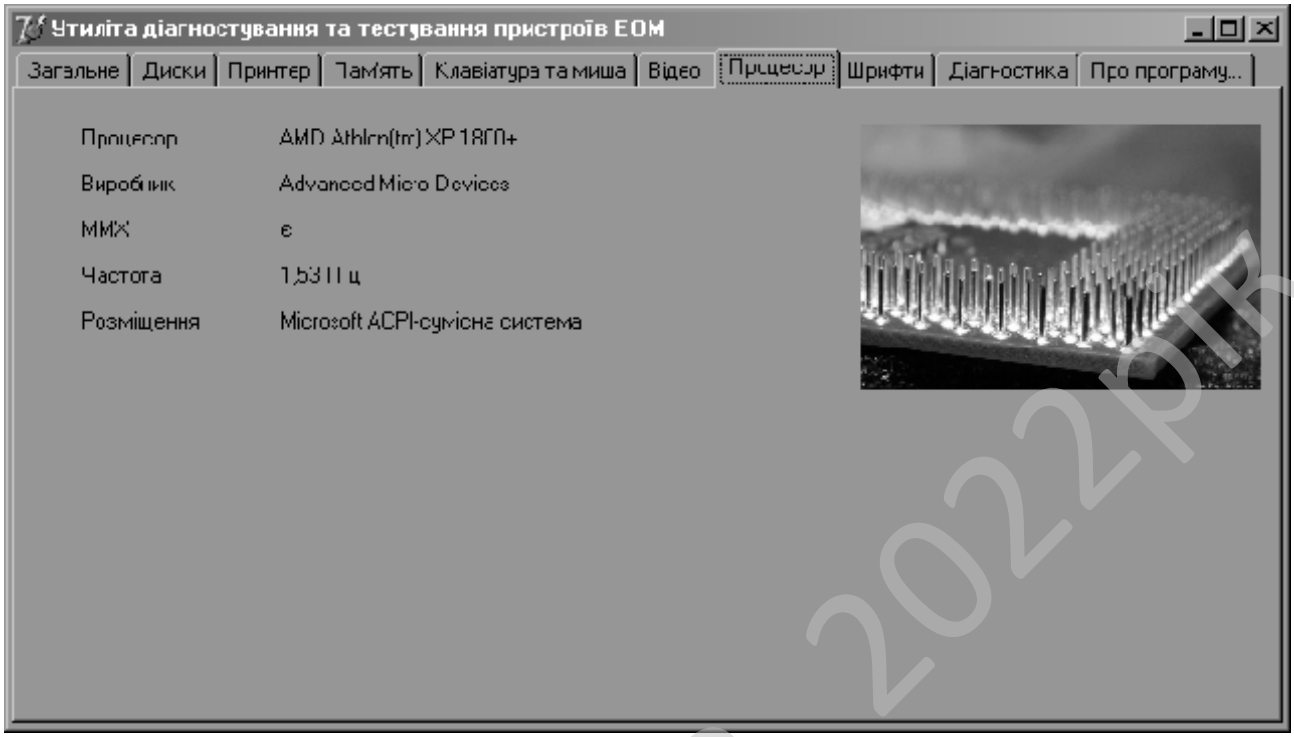

Рисунок 5.8 – Закладка «Інформація про процесор»

Шрифти – інформація про встановлені шрифти.

Рисунок 5.9 – Закладка «Шрифти»

Кафедра_КбПЗ_2022рік

Діагностика – тут зібрані такі властивості, як тестування процесора, пам'яті й перевірка лотка CD/DVD-rom, а також інформація про автора цієї програми.

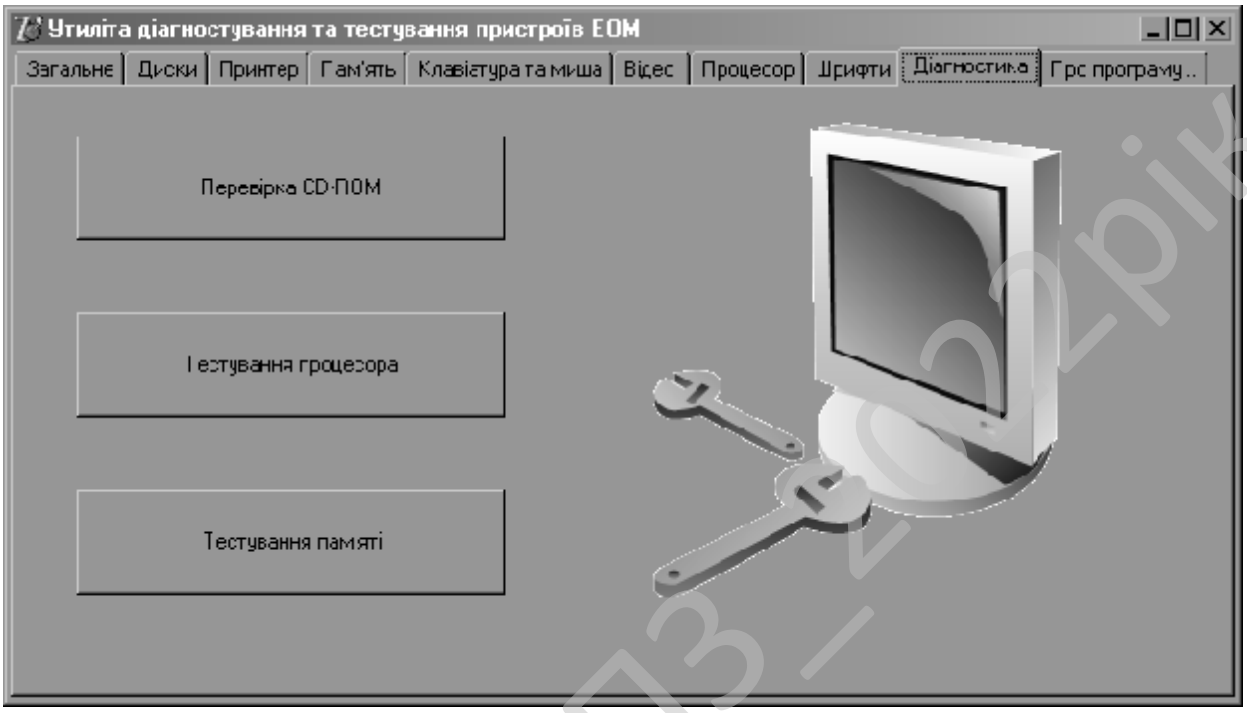

Рисунок 5.10 – Закладка «Діагностика»

При натисканні на кнопку тестування процесора випадає вікно зображене на рисунку 5.11, де відбувається тестування процесора.

Тут, можна протестувати дуже багато параметрів. Перелічимо найбільш головні з них:

– тактова частота – тут застосовується убудований у мікропроцесор лічильник тактів, з якого ми одержуємо поточну тактову частоту;

– швидкість виконання арифметичних операцій – тут ми тестуємо, наскільки швидко процесор виконує основні арифметичні операції з різними типами даних;

– швидкість системної шини – тут ми перевіряємо швидкість шини між процесором і пам'яттю, тобто за скільки тактів передається інформація між ними. Використовується інструкції НОП, розташовані в першому випадку в оперативній пам'яті, а в другому випадку – у кеш-пам'яті.

Кафедра_КОПЗ_2022рік

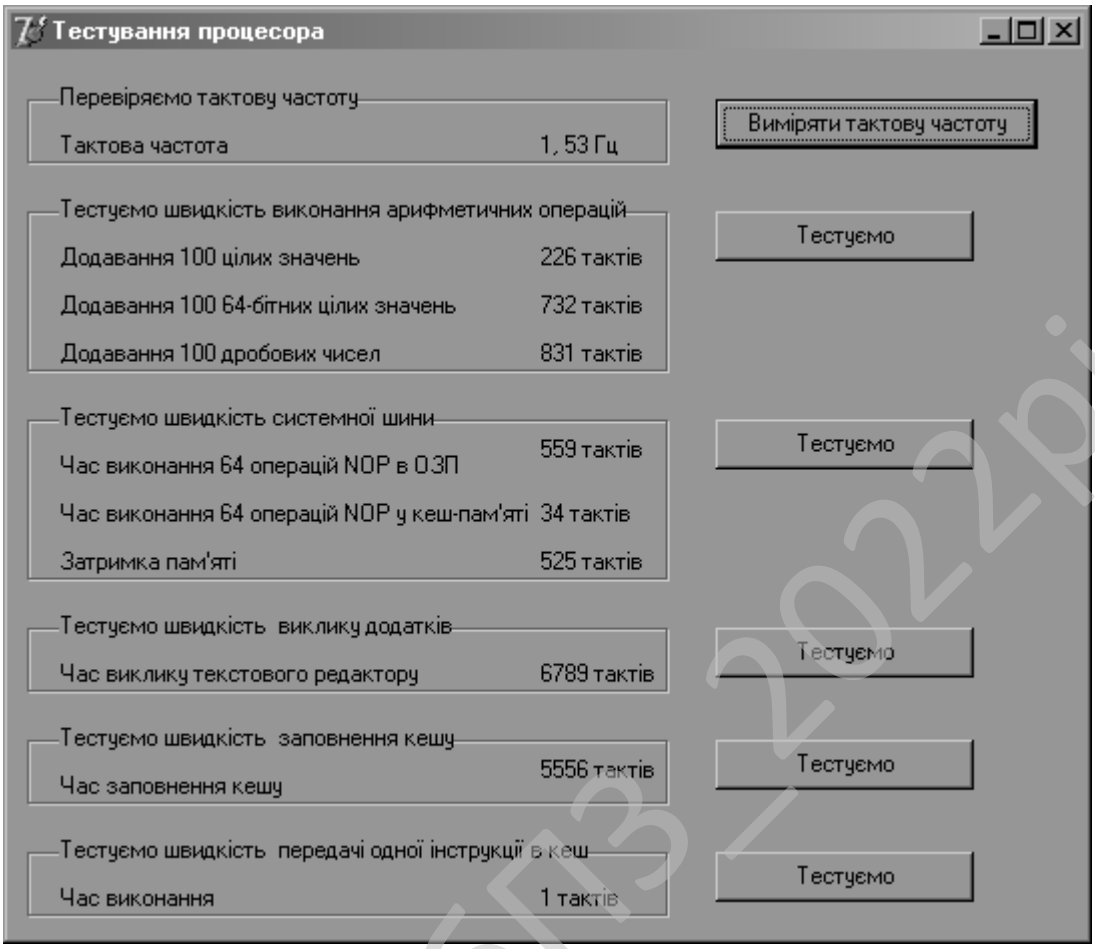

Рисунок 5.11 – Закладка «Тестування ЦП»

Наприкінці програма обчислює затримку пам'яті, тобто скільки тактів потрібно саме на передачу даних з/до пам’яті:

– швидкість виклику додатків – тут тестуємо, за скільки тактів викликається в оперативну пам'ять стандартна програма Блокнот

– швидкість заповнення кешу – там ми заповнюємо всі 256 або 512 кілобайт кешу інструкціями, і дивимося, за скільки тактів він заповниться.

Коротку довідку про програму можна переглянути натиснувши на закладку «Про програму…» (рисунок 5.12).

Кафедра КПЗ 2022рік

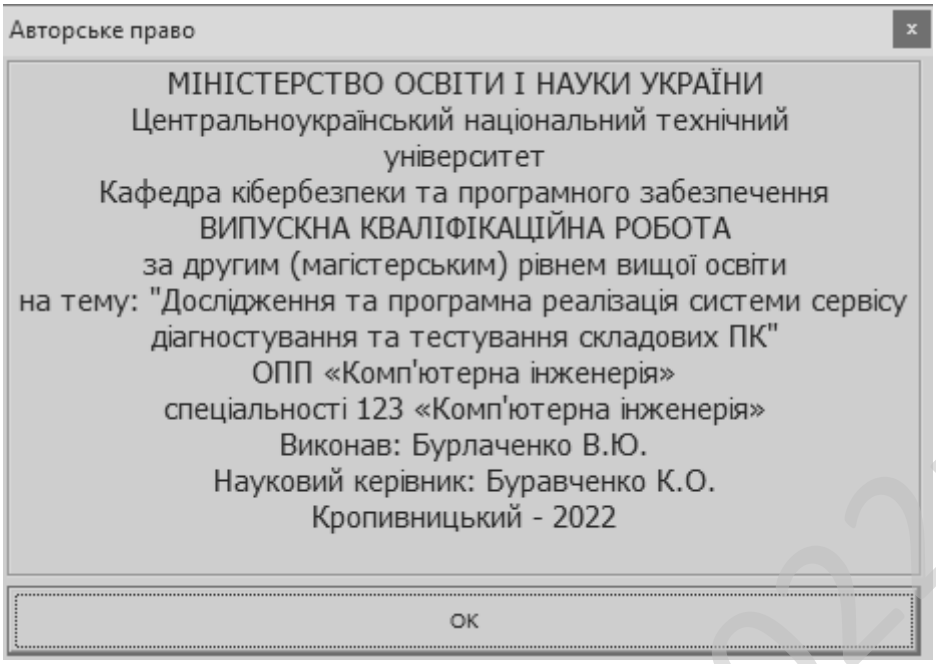

Рисунок 5.12 – Закладка «Про програму…»

Кафедра_КбПЗ_2022рік

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи сервісу діагностування та тестування складових ПК.

Метою розробки є дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК.

Об'єктом дослідження є процес сервісу діагностування та тестування складових ПК.

Предметом дослідження є методи сервісу діагностування та тестування складових ПК.

Методи дослідження базуються на методах схемотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод сервісу діагностування та тестування складових ПК.

– Розроблено вітчизняний продукт сервісу діагностування та тестування складових ПК, який має більш широкі можливості, на відміну від існуючих аналогів.

Кафедра_КБПЗ_2022рік

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи сервісу діагностування та тестування складових ПК. Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

а) невеликий розмір;

б) невеликі системні потреби;

в) незалежність від встановлених на комп'ютері баз даних;

г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	24
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Кафедра_КбПЗ_2022рік

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Кафедра КбПЗ 2022рік

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	24000
33. Норматив додаткової зарплати, % :	$Н_д$	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A\ \text{Size}^B, \qquad (7.1)$$

де: A – коефіцієнт Боема, A = 2,45;

Кафедра_КбПЗ_2022рік

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1{,}01 + 0{,}001\sum W_i, \qquad (7.2)$$

де: Wi – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1{,}01+0{,}001(2{,}43+3{,}64+3{,}38+3{,}95+2{,}73) = 1{,}027.$$

$$T_{ном} = 2{,}45 \cdot 2{,}7^{1{,}026} = 6{,}78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном}\, ПВ_j, \qquad (7.3)$$

де: $ПВ_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6{,}78 \cdot (0{,}88 \cdot 0{,}93 \cdot 0{,}88 \cdot 0{,}91 \cdot 0{,}95 \cdot 1 \cdot 1 \cdot 0{,}87 \cdot 1{,}22 \cdot 1{,}16 \cdot 1{,}1 \cdot 1{,}1 \cdot$$
$$\cdot 1{,}12 \cdot 1{,}1 \cdot 1 \cdot 1{,}1 \cdot 1{,}1) = 9{,}37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0{,}3 C T_{уточн}^{0{,}33+0{,}2(B-1{,}01)} S, \qquad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25…350)%.

$$T_{РП} = 0{,}3 \cdot 2{,}66 \cdot 9{,}37^{0{,}33+0{,}2(1{,}026-1{,}01)} \cdot 58 = 98 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

Кафедра_КПЗ_2022рік

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	98	Ф 7.1-7.4
Впровадження	13	Д13
Всього	139	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{пз} N}{F_{рq} - H_{ев}}, \qquad (7.5)$$

де: $F_{рq}$ – плановий фонд робочого часу одного спеціаліста, днів;

$T_{пз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{139 \cdot 1}{60 - 5} = 2,5 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Кафедра_КПЗ_2022рік

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість облад-нання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	7	630	10,5
Монітор	60	7	420	7
Клавіатура	30	7	210	3,5
Маніпулятор «мишка»	30	7	210	3,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м.п.	2,5	200	500	8,33
Копіювальний апарат	140	1	140	2,33
Усього за рік:			$З_ч$	39,49

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків неповинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$Ф_{др}^{с} = \frac{З_ч \cdot n_{тіс}}{1,2}, \quad (7.6)$$

$$Ф_{др}^{с} = \frac{39 \cdot 3}{1,2} = 97,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{Ф_{др}^{с}}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

Кафедра_КОПЗ_2022рік

$$Ч_{ел} = 97,5/(60 \cdot 8) = 0,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпровідного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Кафедра_КБПЗ_2022рік

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Кафедра_КОПЗ_2022рік

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	19038	57114
Продакт-менеджер	0,25	18000	13500
Інженер-програміст	2,5	19000	142500
Інженер-електронщик	0,2	16500	9900
Інженер-системотехнік	0,25	16500	12375
Адміністратор мережі	0,5	16500	24750
Системний програміст	0,25	16500	12375
Дизайнер WEB	0,25	16500	12375
Інженер-верстальник	0,25	16500	12375
Бухгалтер-економіст	0,5	16500	24750
Всього за період розробки	$R_{сп}$ = 5,95	-	$Ф_{роб}$ = 322014

Розрахуємо середньоденну зарплату одного виконавця:

$$З_{сд} = \frac{Ф_{роб}}{R_{сп} F_{рд}}, \quad (7.8)$$

де: $Ф_{роб}$ – загальна сума зарплати за плановий період, грн.

$$З_{сд} = \frac{322014}{5,95 \cdot 60} = 902 \ грн.$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

Кафедра_КбПЗ_2022рік

$$Б_{уд} = R^1_{сп} S_у Ц_{пл}, \quad (7.9)$$

де: $R^1_{сп}$ – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

$S_у$ – питома площа на одне робоче місце, $м^2$;

$Ц_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних інтернет ресурсу DOM.RIA (https://dom.ria.com) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500…1600 $у.о./м^2$. Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./м^2. На кожне робоче місце у середньому потрібно 8 $м^2$. З урахуванням цього:

$$Б_{уд} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$І_{нв} = R^1_{сп} \cdot Ц_м, \quad (7.10)$$

де: $Ц_м$ – ціна меблів для одного робочого місця, грн.

$$І_{нв} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом фірми Brain за 24.10.22 – джерело http: //brain.com.ua.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771

Кафедра_КБПЗ_2022рік

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Системний блок		7771
Процесор	AMD A8-3870K 3.0 GHz Socket FM1 Qua Core	2500
Системна плата	ASUS F1A55-M , AMD A55 , Socket FM1	1100
Жорсткий диск	HDD 500 Gb SAMSUNG Barracud HD502HJ (3.5", 500ГБ, 16МБ, SATA II-300)	1290
Оперативна пам’ять	DIMM 4096Mb DDR3 PC3-12800 Kingstor 1600MHz, 512M x 64, CL9-9-9-27, 1.65V w/heatsink, HyperX	834
DVD-привод	DVD±RW ASUS DRW-24B5ST Black Bulk	416
Корпус	Logicpower 8702 – 550w 12cm	1411
Кардрідер внутрішній	Transcend TS-RDF8K USB 3.0	220
інше	Клавіатура, мишка	Подаруно
Монітор	Монітор BenQ GL2450HM Black	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USI cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійног живлення	Powercom BNT-600AP USB	1400

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Кафедра_КОТЗ_2022рік

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	8	11771	9416,8	103584,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	125216,3

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400

Кафедра_КБПЗ_2022рік

Продовження таблиці 7.8

1	2	3	4
Група 4			
3. Обчислювальна техніка	125216	-	-
Всього по групі	125216	50	62608
Група 5,6			
4. Вимірювальні пристрої	5190	20	-
5. Транспортні засоби	143000	25	
6. Господарський інвентар	28000	20	-
Всього по групам 5, 6	176190	-	35238
7. Нематеріальні активи	24000	10	2400
Разом	$К_р$= 1733406		$А_р$= 170646

Примітка: вартість автомобіля Volkswagen Fox 2005 взята за даним сайту «Авто-РІА», джерело https://auto.ria.com/uk/auto_volkswagen_fox_33599812.html, складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$З_о = \frac{З_{сд} \cdot Т_{пз}}{N_е}, \quad (7.11)$$

де: $N_е$ – кількість екземплярів програм, шт.

$$З_о = 902 \cdot 139/24 = 5225 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

Кафедра_КбГЗ_2022рік

$$З_д = З_о \cdot Н_q \cdot 0{,}01, \quad (7.12)$$

де: $Н_q$ – норматив додаткової зарплати, %.

$$З_д = 5225 \cdot 10 \cdot 0{,}01 = 523 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $Н_с = 22\%$ від суми основної та додаткової зарплати:

$$С_{оц} = 0{,}01 \cdot Н_с (З_о + З_д), \quad (7.13)$$

де: $Н_с$ – відрахування на соціальні потреби, %.

$$С_{оц} = 0{,}01 \cdot 22(5225+523) = 1265 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $Н_г = 15\%$ від основної зарплати:

$$Г_{осп} = З_о \cdot Н_г \cdot 0{,}01, \quad (7.14)$$

де: $Н_г$ – загальногосподарські витрати, %.

$$Г_{осп} = 5225 \cdot 15 \cdot 0{,}01 = 784 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$З_М = (З_{М1} + З_{М2} + З_{М3})/N_е, \quad (7.15)$$

де: $З_{М1}$ – вартість паперу, грн.;

$З_{М2}$ – вартість запам’ятовуючих пристроїв, грн.;

$З_{М3}$ – вартість фарби, картріджей, тонеру, грн.;

$N_е$ – кількість екземплярів програм, шт.

Згідно прийнятих норм на підприємстві $n_{вит}$ приймаємо 1,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_п$=210 грн., визначаємо вартість паперу за період розробки:

$$З_{М1} = Ц_п \cdot N_м. \quad (7.16)$$

$$З_{М1} = 210 \cdot 1{,}5 = 315 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам’ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 10):

Кафедра_КОПЗ_2022рік

$$З_{М2} = \sum Ц_д \,, \tag{7.17}$$

де: $Ц_д$ – вартість дисків CD/DVD: CDR box – 23,6 грн./шт., DVD-R box – 30 грн./шт.

$$З_{М2} = 30 \cdot 10 = 300 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$З_{М3} = \sum Ц_з ., \tag{7.18}$$

де: $Ц_з$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$З_{М3} = 574+558+570 = 1702 \text{ грн.}$$

$$З_М = (315 + 300 + 1702)/24 = 97 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($Н_п$ = 15%) від основної зарплати виконавців:

$$О_п = З_о \cdot Н_п \cdot 0{,}01, \tag{7.19}$$

де: $Н_п$ – норматив витрат на освоєння нових мов програмування, %.

$$О_п = 5225 \cdot 15 \cdot 0{,}01 = 784 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_е$ = 24 прим.):

$$А_м = \frac{А_р \cdot N_{міс}}{N_е \cdot 12}, \tag{7.20}$$

де: $А_р$ – загальна річна сума амортизаційних відрахувань, грн.

$$А_м = 170646 \cdot 3/(24 \cdot 12) = 1777 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$С_п = З_о + З_д + С_{оц} + Г_{осп} + З_м + О_п + А_м \,. \tag{7.21}$$

$$С_п = 5225+523+1265+784+97+784+1777 = 10455 \text{ грн.}$$

Кафедра_КОПЗ_2022рік

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$П_р = 0{,}01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$П_р = 0{,}01 \cdot 50 \cdot 10455 = 5228 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	$З_о$	5225
2. Додаткова зарплата виконавців	$З_д$	523
3. Відрахування на соціальні потреби	$С_{оц}$	1265
4. Загальногосподарські витрати	$Г_{осп}$	784
5. Витрати на матеріали	$З_М$	97
6. Освоєння нових операційних систем, мов програмування	$О_п$	784
7. Амортизація основних фондів	$А_м$	1777
8. Повна собівартість програмного забезпечення	$С_п$	10455
9. Плановий прибуток	$П_р$	5228
10.Ціна підприємства $Ц_п = С_п + П_р$	$Ц_п$	15683
11.Податок на додану вартість $ПДВ = 0.01 \cdot Н_{дв} \cdot Ц_п$	$ПДВ$	3136,6
12.Відпускна ціна програмної продукції $Ц = Ц_п + ПДВ$	$Ц$	18819,6

Кафедра_КОПЗ_2022рік

7.6 Визначення об’єму капітальних вкладень у споживача програмної продукції

Об’єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконалагоджувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об’єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	18820
Всього капітальних витрат	–	18820

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

$$З_о = Ч_е О_е М_е (1 + 0{,}01 Н_q)(1 + 0{,}01 Н_c), \qquad (7.23)$$

де $Ч_е$ – чисельність працівників, чол.;

$О_е$ – заробітна плата, грн./год;

$М_е$ – час, що витрачається на нарахування ЗП.

Після купівлі нового програмного забезпечення час на нарахування зменшився з 630 годин на рік до 393 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$З_{о\ баз} = 1 \cdot 63 \cdot 630 \cdot 1{,}1 \cdot 1{,}22 = 53264\ грн,$$

до:

$$З_{о\ нов} = 1 \cdot 63 \cdot 393 \cdot 1{,}1 \cdot 1{,}22 = 33227\ грн.$$

Кафедра_КОПЗ_2022рік

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Заробітна плата(основна, додаткова, відрахування в соціальні фонди)	$З_р$	53264	33227
2. Витрати на електроенергію	$З_{ел}$	9639	8925
3. Витрати на амортизацію	$З_{ам}$	0	4705
Всього витрат за рік	$І$	62903	46857

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($П_{ел}$) в кіловатах, часу експлуатації технічних засобів ($Т_р$) в годинах та ціни однієї кіловат-години ($Ц_{ел}$):

$$З_{ел} = П_{ел} \cdot Т_р \cdot Ц_{ел}. \quad (7.24)$$

$$З_{ел\ баз} = 0{,}54 \cdot 8500 \cdot 2{,}1 = 9639 \text{ грн.}$$

$$З_{ел\ нов} = 0{,}50 \cdot 8500 \cdot 2{,}1 = 8925 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	18820	–	4705
Всього відрахувань	-	–	18820	–	4705

Кафедра_КПЗ_2022рік

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_в = (Ц_п - C_п) \cdot N_е - \sum_{i=1}^{m} E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: $K_р$ – балансова вартість основних фондів розробника, грн.; $E_р$ – розрахунковий коефіцієнт капіталовкладень.

$$Ев = (15683 - 10455) \cdot 24 - (0,05 \cdot 1408000 + 0,5 \cdot 125216 +$$
$$+0,25 \cdot 33190 + 0,2 \cdot 143000 + 0,1 \cdot 24000) \cdot 3/12 = 82396 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_в = \frac{K_p^*}{(Ц_п - C_п) \cdot N_е}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_в = \frac{325406}{(15683 - 10455) \cdot 24 \cdot 12 / 3} = 0,7 \text{ років}.$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{сп} = (I_б - I_н) - E_н(K_н - K_б), \quad (7.27)$$

де: $I_б$, $I_н$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_б$, $K_н$ – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{сп} = (62903\text{-}46857)\text{-}0,25 \cdot 18820 = 11341 \text{ грн.}$$

Кафедра_КбПЗ_2022рік

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	24
2. Повна собівартість розробленої програми	Грн.	10455
3. Ціна розробленої програми	Грн.	15683
4. Плановий прибуток від реалізації розробленої програми	Грн.	5228
5. Рентабельність програмної продукції	%	50
6. Об’єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1733406
7. Загальний прибуток від реалізації програмної продукції	Грн.	125472
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	82396
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,7
10.Об’єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	18820
11.Величина економічного ефекту у користувача програмної продукції	Грн.	11341
12.Період окупності додаткових капітальних вкладень у користувача програмної продукції	Рік	1,2

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{сп} = \frac{K_н - K_б}{I_б - I_н}, \qquad (7.28)$$

Кафедра_КбПЗ_2022рік

$$T_{сп} = \frac{18820}{62903 - 46857} = 1,2 \text{ років}.$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

Кафедра_КбПЗ_2022рік

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1. Вступ

Аналізуючи умови працівників it-сфери, на перший погляд, може здатися, що працівники сфери інформаційних технологій не схильні до ризиків на виробництві, та якщо більш глибоко розглянути умови і специфіку праці фахівців сфері it-індустрії, можна виявити ряд факторів які будуть мати негативний сплив на стан охорони праці, та на самого it-фахівця зокрема. Сюди можна віднести як невідповідність освітлення, так і високий рівень шуму, що негативно позначатимуться як на емоційному так і на фізичному стані фахівця, призводитимуть до зниження ефективності праці та виробничих травм. Також, важливим моментом охорони праці it-фахівця є врахування його психологічних можливостей (швидкість реакції, особливості пам'яті та уваги, емоційний стан, тощо). Для того, щоб забезпечити ефективну роботу it-фахівця, потрібно враховувати та максимально компенсувати такі негативні фактои як: надмірне нервово-емоційне навантаження, довготривалі статичні перевантаження, обмежена рухова активність. Всі ці чинники призводить до різноманітних відхилень у стані здоров'я, зокрема до перевтоми, зниження фізичної та розумової працездатності, неврозів, захворювань серцево-судинної системи тощо. Метою даного розділу є огляд конкретних умов праці спеціаліста у сфері it-індустрії. Завданнями для даного розділу є: аналіз умов праці на робочому місці фахівця it-індустрії, розробка конкретних рекомендацій щодо покрашення умов праці фахівців it-індустрії, огляд пожежної безпеки на it-підприємстві та розрахунок системи загального штучного освітлення виробничого приміщення де працюють IT – фахівці.

Кафедра_КОПЗ_2022рік

8.2. Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

– підвищений рівень шуму;

– несприятливі мікрокліматичні умови;

– недостатній рівень освітленості;

– шкідливі речовини;

– підвищений рівень електромагнітних випромінювань радіочастот;

– висока напруга електричної мережі;

– статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним

Кафедра_КБПЗ_2022рік

напруженням) категорій. В таблиці 8.1. наведені оптимальні параметри мікроклімату в приміщеннях.

Таблиця 8.1 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	22...24°С; 40... 60%; до 0,1 м/с
Теплий	Температура повітря в приміщенні; відносна вологість; швидкість руху повітря	23...25 °С 40...60% 0,1...0,2 м/с

Виміряні за допомогою приладів температура та вологість у приміщеннях праці ІТ-фахівців повинні відповідати зазначеним у таблиці для теплого періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціювання повітря, або забезпечити подачу свіжого повітря системами вентиляції. Норми подачі свіжого повітря наведені у таблиці 8.2.

Таблиця 8.2 – Норми подачі свіжого повітря в приміщення

Характеристика приміщення	Об'ємна витрата свіжого повітря, що подається в приміщення, $м^3$ на одну людину в годину
Об'єм до 20 $м^3$ на людину 20... 40 $м^3$ на людину Більше 40 $м^3$на людину	Не менше 30 Не менше 20 Може біти використана природна вентиляція

Кафедра_КПЗ_2022рік

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і великого зосередження, забарвлення повинно бути спокійних тонів – малонасичені відтінки холодного зеленого або блакитного кольорів.

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Раціональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [2]

8.3 Пропозиції щодо підвищення працездатності ІТ – фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців it-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців it-індустрії і стандартів підприємств,

Кафедра_КОПЗ_2022рік

центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальний працездатності і відновленню психосоматичного здоров'я фахівців it-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців it-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Для більшого розуміння, пропозиції щодо підвищення працездатності it-фахівців, розіб'ємо на декілька категорій:

1 Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням it-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють it-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження it-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці it-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

2 Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність it-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві it-галузі. Тому нами пропонується закупівля тільки меблів , які

Кафедра_КОПЗ_2022рік

пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тімбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

8.4 Розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці

Приміщення з ПК повинні мати природне і штучне освітлення, яке відповідало б вимогам ДБН В.2.5-28-2006 «Природне і штучне освітлення», ДСАНПІН 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин». Приміщення для роботи із ПЕОМ повинні мати природне й штучне освітлення. Віконні прорізи повинні бути орієнтовані на північ або на північний схід, забезпечувати коефіцієнт природної освітленості (К.П.О.) не менш 1,5% і мати жалюзі або штори. Віконні прорізи повинні мати регульовані пристрої для відкривання, а також жалюзі, завіски, зовнішні козирки тощо. Приміщення із ПЕОМ повинні бути обладнані системою загального рівномірного освітлення. У виробничих і адміністративно-суспільних 130 приміщеннях, де переважно ведеться робота з документами, допускається комбінована система штучного освітлення. Штучне освітлення має здійснюватися системою загального рівномірного освітлення, яка включає суцільні або такі, що перериваються лінії світильників, розташованих збоку робочих місць (переважно ліворуч), паралельно лінії зору користувачів ПК. Світильники повинні мати розсіювачі світла. У світильниках місцевого освітлення можна використовувати лампи накалювання. При розміщенні ПК по

Кафедра_КОПЗ_2022рік

периметру приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями. Система освітлення робочого місця користувача ПК має відповідати наступним вимогам (рис. 8.1).

Рисунок 8.1 – Вимоги до системи освітлення робочого місця користувача ПК

Освітленість на робочому столі користувача в зоні розташування документів має бути в межах 300-500 лк. Якщо цей рівень освітленості неможливо забезпечити системою загального освітлення то допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану (яскравість відблисків не повинна перевищувати 40 кд/м^2) та перевищення його освітленості більше ніж 300 лк. Яскравість світильників загального освітлення, а також яскравість стелі при застосуванні системи відбитого освітлення не повинна перевищувати 200 кд/м^2. Величина

Кафедра_КОП_2022рік

коефіцієнта пульсації освітленості не повинна перевищувати 5%. Що стосується розподілу яскравості в полі зору працюючих за дисплеями ПК, то відношення значень яскравості робочих поверхонь не повинно перевищувати 3:1

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 6 м, довжина – 7 м, висота – 2,9 м. У зазначеному приміщенні працює 7 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою [1]:

$$F=ESKZ/n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; *E = 300 Лк;*

S – площа освітлюваного приміщення (у нашому випадку *S*=6 х 7 = 42 $м^2$);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку *K = 1,5);*

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку *Z = 1,1*);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці [8]; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін **($\rho_{стін.}$)** і стелі **($\rho_{стелі}$),** значення коефіцієнтів дорівнюють $\rho_{стін}$ **= 50% і** $\rho_{стелі}$ **= 50%.**

Обчислимо індекс приміщення за формулою:

$$i=S/(h(A+B)),$$

Кафедра_КБПЗ_2022рік

де:

S – площа приміщення, *S* = 42 *м*^{2}*;*

h – розрахункова висота підвісу, *h* = 2,9 *м (співпадає з в*исотою стелі, т.я. лампи освітлення закріплюються на стелі)*;*

A – ширина приміщення, *A = 6 м;*

B – довжина приміщення, *B =* 7 *м.*

Підставимо всі значення у формулу та визначемо індекса приміщення:

i=1,4.

Знаючи індекс приміщення, за знаходимо n = 0,29 (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам) [*8]*. Підставимо всі значення у формулу, визначемо світловий потік: *F=71689 Лм.*

Для розрахунку дудемо використовувати *світлодіодні стельові панелі Delux LED Panel 41 44Bm.*, світловий потік яких $F_л$ *= 3600* Лм.

Число ламп визначається по формулі:

$$N=F/F_л$$

де:

F – світловий потік,

$F_л$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначемо індекса приміщення: *N= 71689 / 3600=19,9 шт.*

Приймаємо необхідну кількість *світлодіодних світильників* 20 шт.

8.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров’я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці на робочому місці ІТ-фахівця, призначеного для праці програмістів, проведено розгляд небезпечних та

Кафедра_КбПЗ_2022рік

шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з умов поліпшення охорони праці.

Кафедра_КбПЗ_2022рік

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи сервісу діагностування та тестування складових ПК.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів сервісу діагностування та тестування складових ПК.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем сервісу діагностування та тестування складових ПК.

– Досліджена система сервісу діагностування та тестування складових ПК.

– На основі отриманих результатів досліджень створена програмна реалізація системи сервісу діагностування та тестування складових ПК.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання сервісу діагностування та тестування складових ПК.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

КафедраКбПЗ_2022рік

При створені програмного забезпечення було використано об’єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Embarcadero RAD Studio Delphi 10.3.2 Rio Architect. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм LOKI_91.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 11341 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,2 роки.

Кафедра_КОПЗ_2022рік

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бурлаченко В.Ю. Дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.

2. Коваленко А.С. Разработка структуры базы данных интегрированной информационной системы / А.С. Коваленко, А.В. Коваленко // Информационные технологии и защита информации в информационно-коммуникационных системах: монографія / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – С. 54-64.

3. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.

4. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко ., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.

5. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко , О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.

6. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

7. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы /

Кафедра_КОПЗ_2022рік

А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

8. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

9. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

10. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.

11. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.

12. Коваленко А.С. Обгрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.

13. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.

14. Коваленко А.С. Удосконалення методу технічного обслуговування об’єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов,

Кафедра_КОПЗ_2022рік

О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.

15. Коваленко А.С. Метод визначення оптимального комплексу робіт з відновлення працездатності інтегрованої системи технічної діагностики в умовах ресурсних обмежень / А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2016. – Вип. 3(140). – С. 69-72.

16. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.

17. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.

18. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.

19. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

20. Коваленко А.С. Задачи распознавания ситуаций в системах организационной стратегии интеграции производства и операций

Кафедра_КБПЗ_2022рік

/ А.С. Коваленко, А.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVI міжнар. наук.-практ. сем., 11-12 квіт. 2014 р., м. Кіровоград: зб. тез. – Кіровоград: КНТУ, 2014. – С. 53-55.

21. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.

22. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.

23. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.

24. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.

25. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

26. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у

Кафедра_КОПЗ_2022рік

промисловості і освіті: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

27. Коваленко А.С. Обґрунтування необхідності створення розподіленої бази даних для забезпечення захисту рухомих повітряних об`єктів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Перспективні напрями захисту інформації: I всеукр. наук.-практ. конф., 07 вер. 2015 р., м. Одеса: зб. тез доп. – Одеса: ОНАЗ, 2015. – С. 35-39.

28. Коваленко А.С. Розробка інформаційної моделі автоматизованої оцінки технічного стану інтегральної інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформаційні технології та взаємодії (IT & I): II між нар. наук.-практ. конф., 3-5 лист. 2015 р., м. Київ: тези доп. – Київ: КНУ ім. Т. Шевченка, 2015. – С. 41-42.

29. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Информационные и телекоммуникационные технологии: образование, наука, практика: II междунар. научн.-практ. конф., 3-4 дек. 2015 г., г. Алматы, Казахстан: сб. труд. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – Т.2. – С. 423-427.

30. Королюк Н.А. Оценка временных интервалов работы лица, принимающего решение, на автоматизированном командном пункте / Н.А. Королюк, А.И. Тимочко // Системи обробки інформації. – Х.: ХУПС, 2005. – Вип. 8 (48). – С. 51-54.

31. Костерев В.В. Надёжность технических систем и управление риском: учебн. пособ. / В.В. Костерев. – М.: МИФИ, 2008. – 280 с.

32. Костюков А.В. Підвищення операційної ефективності підприємств на основі моніторингу в реальному часі. / А.В. Костюков, В.М. Костюков. – М.: Машинобудування, 2009. – 192 с.

33. Лазарев А.А. Выбор показателя затрат для анализа сравнительной экономической эффективности техники конечного потребления / А.А. Лазарев,

Кафедра_КОПЗ_2022рік

М.В. Бейлин // Сборник научных трудов ХГПУ.– Х.: ХГПУ, 1999. – Вып. 74. – С. 27-29.

34. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 1. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 308 с.

35. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 2. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 208 с.

36. Лапсарь А.П. Метод оценки состояния сложных технических объектов для синтеза быстродействующих прогнозирующих систем / А.П. Лапсарь // Измерительная техника. – 2004. – № 2. – С. 7-10.

37. Линейные задачи оптимизации: Учеб. пособие / С.В. Лутманов. – Пермь: ЛИТЕР-А, 2004. – Ч.1. – Линейное программирование. – 128 с.

38. Литвак Б.Г. Экспертные технологии в управлении. Учебное пособие / Б.Г. Литвак. – М.: Дело, 2014. – 318 с.

39. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: Посібн. / В.М. Локазюк, Ю.Г. Савченко. – К.: Видавничий центр «Академія», 2004. – 376 с.

40. Лопатников Л. И. Экономико-математический словарь: Словарь современной экономической науки / Л.И. Лопатников. – М.: Дело, 2003. – 520 с.

41. Манухина С.Ю. Инженерная психология и эргономика: хрестоматия / С.Ю Манухина. – М.: Изд. центр ЕАОИ, 2009. –224 с.

42. Мартыненко М.В. Человекомашинные процедуры поддержки организационно–управленческих решений: учеб. пособие СПбГЭТУ / М.В. Мартыненко, О.И. Шеховцов. – СПб, 2012. – 250 с.

43. Мунипов О.В. Эргономика: человекоориентированное проектирование техники, программных средств и среды: Учебник / О.В. Мунипов, В.П. Зинченко. – М.: Логос, 2001. – 356 с.

Кафедра_КОПЗ_2022рік

44. Надеев А.И. Математическая модель эксплуатационной надежности интеллектуальных датчиков / А.И. Надеев, Р.А. Юсупов, Ю.К. Свечников, Д.Р. Юсупов // Измерительная техника. – М: Стандартинформ, 2004. – № 1. – С. 8-11.

45. Надійність техніки. Аналіз надійності. Основні положення: ДСТУ 2861-94 – [Чинний від 1997–01–01]. – Київ: Держстандарт України, 1995. – 33 с. – (Національний стандарт України).

46. Надійність техніки. Терміни та визначення: ДСТУ 2860-94 – [Чинний від 1996–01–01]. – Київ: Держстандарт України, 1994. – 36 с. – (Національний стандарт України).

47. Нейлор К. Как построить свою экспертную систему / К. Нейлор. – М.: Энегроатомиздат, 2007. – 242 с.

48. Николаева И. П. Экономический словарь / И.П. Николаева. – Проспект, 2015. – 399 с.

49. Онищук А.Г. Радиоприемные устройстваю: Учебн. пособ. – 2-е изд., испр. / А.Г. Онищук, И.И. Забеньков, А.М. Амелин. – Минск: Новое знание, 2007. – 240 с.

50. Осипов В. Базы данных и Delphi. Теория и практика / В. Осипов. – БХВ-Петербург, 2011. – 752 с.

51. Павленко М.А. Метод кольорового кодування інформаційних елементів при розробці інформаційних моделей в перспективних АСУ / М.А. Павленко, П.Г. Берднік, Д.В. Прибильнов // Наукова весна – 2008: Матеріали міжнародної наук. – практ. конф. – Х.: МСУ, 2008. – С. 25-27.

52. Павленко М.А. Метод разработки модели деятельности оператора АСУ в системах управления сложными динамическими объектами / М.А. Павленко, О.С. Бодяк, М.Ю. Гусак, С.И. Симонов // Системи обробки інформації. – Харків: ХУПС, 2012. – Вип. 9(107). – С. 196-200.

53. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-

Кафедра_КОПЗ_2022рік

007-98. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/v0007282-98

54. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-XII. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/2694-12

55. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

56. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/z0508

57. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/handle/123456789/4358

58. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/va042282-99

59. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : "Колос", 1973. – 238 с.

60. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: https://cpo.stu.cn.ua

61. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/handle/123456789/12161 (дата звернення 19.09.22).

Кафедра_КОПЗ_2022рік

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....2
2 Підстава для розробки.....2
3 Мета та призначення розробки.....2
4 Джерела розробки.....2
5 Технічні вимоги.....2
5.1 Вміст проекту.....2
5.2 Показники призначення.....3
5.3 Вимоги до функціональних характеристик.....3
5.4 Вимоги до архітектури.....3
5.5 Вимоги до надійності.....3
5.6 Умови експлуатації.....4
5.7 Вимоги до складу та параметрів технічних засобів.....4
5.8 Вимоги до інформаційної і програмної сумісності.....4
5.8.1 Обладнання.....4
5.8.2 Мова програмування.....4
5.8.3 Вхідні дані.....5
5.8.4 Вихідні дані.....5
6 Вимоги до програмної документації.....5
7 Економічні вимоги.....5
8 Вимоги щодо охорони праці.....5
9 Перелік документів, що розробляються.....6
10 Етапи розробки.....6
11 Порядок контролю та приймання.....6

Кафедра_КОПЗ_2022рік

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи сервісу діагностування та тестування складових ПК.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 17.08.2022 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

– вибір і обґрунтування методів реалізації проекту;

– розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

Кафедра_КБПЗ_2022рік

– техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

– аналіз умов праці;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– програмну реалізацію системи сервісу діагностування та тестування складових ПК;

– цілісність даних у процесі роботи та при зберіганні;

– простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

Кафедра_КбПЗ_2022рік

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– температура повітря: 19-20 град. по Цельсію;

– відносна вологість повітря до 80%;

– атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпечення.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп’ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14” 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Embarcadero RAD Studio Delphi 10.3.2 Rio Architect.

Кафедра_КОПЗ_2022рік

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці на робочому місці ІТ-фахівця.

Кафедра_КБПЗ_2022рік

9 Перелік документів, що розробляються

– Наукова новизна – 1 аркуш.
– Структурна схема системи – 1 аркуш.
– Функціональна схема системи – 1 аркуш.
– Діаграма процесів – 1 аркуш.
– Блок-схема алгоритму роботи програми – 2 аркуша.
– Показники економічної ефективності – 1 аркуш.
– Пояснювальна записка – 107 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 19.12.2022 р.

Кафедра_КбПЗ_2022рік

Додаток Б

(обов’язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за

другим (магістерським) рівнем вищої освіти

_____ Буравченко К.О.

Дослідження та програмна реалізація

системи сервісу діагностування та тестування складових ПК

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 21

Літера: РП

Кропивницький – 2022 року

Кафедра_КбПЗ_2022рік

Головний модуль

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls;

type
  TForm11 = class(TForm)
    Image1: TImage;
    Timer1: TTimer;
    Label1: TLabel;
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form11: TForm11;

implementation

uses Diag;

{$R *.dfm}

procedure TForm11.Timer1Timer(Sender: TObject);
begin
 diadnostic.show;
 timer1.Enabled:=false;
end;

end.
```

Кафедра_КбПЗ_2022рік

Модуль діагностики

```
unit Diag;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  StdCtrls, ComCtrls, Registry,Printers, ExtCtrls, AxCtrls, OleCtrls,
vcf1, Tabs, Winspool,
  FileCtrl, ImgList, Menus,winsock,ScktComp, Systeminfo,mmsystem,
Buttons,shellapi;
type
  TDiadnostic = class(TForm)
    SysInfo1: TSysInfo;
    Timer1: TTimer;
    Button1: TButton;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    GroupBox3: TGroupBox;
    About: TButton;
    procedure AboutClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure aClick(Sender: TObject);
    procedure disknameClick(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure disknameChange(Sender: TObject);
    procedure ListBox1DrawItem(Control: TWinControl; Index: Integer;
      Rect: TRect; State: TOwnerDrawState);
    procedure ListBox1MeasureItem(Control: TWinControl; Index: Integer;
      var Height: Integer);
    procedure ListBox1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Timer1Timer(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
  private

    { Private declarations }
  public

    { Public declarations }
  end;
var
  Diadnostic: TDiadnostic;

implementation

uses tlhelp32, about, example;
{$R *.DFM}

function GetRootDir:string; external 'Ulandll.dll' index 1;

function getboottype:string; external 'UlanDll.dll';// index 31;

procedure TDiadnostic.AboutClick(Sender: TObject);
begin
form2.show;
end;

procedure GetPrName(processor1:Tlabel);
var SI:TsystemInfo;
begin
GetSystemInfo(SI);
Case SI.dwProcessorType of
386:Processor1.caption:='386';
486:Processor1.caption:='486';
```

Кафедра_КОПЗ_2022рік

```
586:Processor1.caption:='586';
686:Processor1.caption:='686';
end;
end;

procedure GetRegInfoWinNT;
var
  Registryv       : TRegistry;
  RegPath         : string;
  sl,sll          : TStrings;
begin

  RegPath := '\HARDWARE\DESCRIPTION\System';
  registryv:=tregistry.Create;
  registryv.rootkey:=HKEY_LOCAL_MACHINE;
  sl := nil;
  try
    registryv.Openkey(RegPath,false);
    diadnostic.Label28.Caption:=(Registry.ReadString('SystemBiosDate'));
    sl:= ReadMultirowKey(Registry,'SystemBiosVersion');
    diadnostic.memo1.Text:=sl.Text;
  except
  end;
  Registryv.Free;
  if Assigned(sl) then sl.Free;
end;

function GetDisplayDevice: string;
var
  lpDisplayDevice: TDisplayDevice;
begin
  lpDisplayDevice.cb := sizeof(lpDisplayDevice);
  EnumDisplayDevices(nil, 0, lpDisplayDevice , 0);
  Result:=lpDisplayDevice.DeviceString;
end;
procedure getinfovideo;
var
  lpDisplayDevice: TDisplayDevice;
  dwFlags: DWORD;
  cc: DWORD;
begin
diadnostic.memo2.Clear;
lpDisplayDevice.cb := sizeof(lpDisplayDevice);
dwFlags := 0;
cc:= 0;
while EnumDisplayDevices(nil, cc, lpDisplayDevice , dwFlags) do
  begin
    Inc(cc);
    diadnostic.memo2.lines.add(lpDisplayDevice.DeviceString);
       {Так само ми побачимо додаткову інформацію в lpDisplayDevice}
  end;
end;
function LocalIP : string;
type
    TaPInAddr = array [0..10] of PInAddr;
    PaPInAddr = ^TaPInAddr;
var
    phe  : PHostEnt;
    pptr : PaPInAddr;
    Buffer : array [0..63] of char;
    I    : Integer;
    GInitData      : TWSADATA;

begin
    WSAStartup($101, GInitData);
    Result := '';
    GetHostName(Buffer, SizeOf(Buffer));
    phe :=GetHostByName(buffer);
    if phe = nil then Exit;
```

Кафедра_КбПЗ_2022рік

```
    pptr := PaPInAddr(Phe^.h_addr_list);
    I := 0;
    while pptr^[I] <> nil do begin
      result:=StrPas(inet_ntoa(pptr^[I]^));
      Inc(I);
    end;
    WSACleanup;
end;

Function GetCPUSpeed: Double;
const
  DelayTime = 500;
var
  TimerHi      : DWORD;
  TimerLo      : DWORD;
  PriorityClass: Integer;
  Priority     : Integer;
begin
  PriorityClass := GetPriorityClass(GetCurrentProcess);
  Priority := GetThreadPriority(GetCurrentThread);
  SetPriorityClass(GetCurrentProcess, REALTIME_PRIORITY_CLASS);
  SetThreadPriority(GetCurrentThread, THREAD_PRIORITY_TIME_CRITICAL);
  Sleep(10);
  asm
    dw 310Fh // rdtsc
    mov TimerLo, eax
    mov TimerHi, edx
  end;
  Sleep(DelayTime);
  asm
    dw 310Fh // rdtsc
    sub eax, TimerLo
    sbb edx, TimerHi
    mov TimerLo, eax
    mov TimerHi, edx
  end;
  SetThreadPriority(GetCurrentThread, Priority);
  SetPriorityClass(GetCurrentProcess, PriorityClass);
  Result := TimerLo / (1000.0 * DelayTime);
end;
 function CheckDriveType(ch:char): String;
var
  DriveLetter: Char;
  DriveType  : UInt;
begin
  DriveLetter := Ch;
  DriveType   := GetDriveType(PChar(DriveLetter + ':\'));
  Case DriveType Of
  0: Result := '?';
  1: Result := 'Path does not exists';
  Drive_Removable: Result := 'Removable';
  Drive_Fixed    : Result := 'Fixed';
  Drive_Remote   : Result := 'Remote';
  Drive_CDROM    : Result := ' CD-ROM';
  Drive_RamDisk  : Result := 'RAMDisk'
  else
  Result := 'Unknown';
 end;
end;
function GettingHWProfileName: String;
var
  pInfo:  TagHW_PROFILE_INFOA;
begin
  GetCurrentHwProfile(pInfo);
  Result := pInfo.szHwProfileName;
end;
procedure TDiadnostic.FormCreate(Sender: TObject);
var OsVerInfo:Tosversioninfo;
    winver,build:string;
```

Кафедра_КбПЗ_2022рік

```
    Disks:byte;
    buffer:array[0..255]of char;
    wd:string;
    sp:array[0..max_ path-1]of char;
    s:string;
    memorystatus:tmemorystatus;
    dwLength:DWORD;          // sizeof(MEMORYSTATUS)
    dwMemoryLoad:DWORD;      // відсотки пам'яті яка використовується
    dwTotalPhys:DWORD ;      // байт фізичної пам'яті
    dwAvailPhys:DWORD   ;    // свободних байт фізичної пам'яті
    dwTotalPageFile:DWORD ; // байт файлу підкачки
    dwAvailPageFile:DWORD  ;// вільних байт файлу підкачки
    dwTotalVirtual:DWORD   ;// використовуємих байт адресного простору
    dwAvailVirtual:DWORD  ; // вільних використовуємих байт
    ktype:integer;
    R:Tregistry;
    R2:Tregistry;
    disk1:integer;
    msgtext:string;
const
monitorregdir:string='\system\currentcontrolset\ENUM\Display\Default_Monit
or';

videordir:string='\System\currentcontrolset\services\class\display\0000';

processordir:string='Hardware\Description\System\Centralprocessor\0';
begin
button2.click;
Label50.Caption:=GettingHWProfileName;
listbox1.items:=screen.fonts;
numofbuttons.caption:=inttostr(getsystemmetrics(sm_cmousebuttons));
if getsystemmetrics(sm_mousepresent)<>0then ismouse.caption:='Наявний'else
ismouse.caption:='Немає';
for disk1:=0 to diskname.items.count-1 do
begin
disk.lines.add(diskname.items[disk1]+'
'+CheckDriveType(diskname.items[disk1][1]));
end;
{монітор та відео}
///////
R:=tregistry.create;
R.RootKey:=HKEY_LOCAL_MACHINE;
R.OpenKey(monitorregdir,false);
monitortype.caption:=R.ReadString('DeviceDesc');
monitormanufacturer.caption:=R.ReadString('Mfg');
monitorid.caption:=r.readstring('HardwareID');
R.OpenKey(videordir,false);
//drvdesc.caption:=r.ReadString('DriverDesc');
driverdate.caption:=r.readstring('DriverDate');
drvprovider.caption:=r.readstring('ProviderName');
driverver.caption:=r.readstring('ver');
r.closekey;
r.closekey;
getinfovideo;
//////
{Версія BIOS}
GetRegInfoWinNT;
{advanced processor info}
R2:=Tregistry.create;
R2.RootKey:=HKEY_LOCAL_MACHINE;
r2.OpenKey(processordir,false);
processorname.caption:=r2.readstring('Identifier');
vident.caption:=r2.readstring('VendorIdentifier');
if not (r2.readstring('MMXIdentifier')='')then
mmx1.caption:=r2.readstring('MMXIdentifier')
else
mmx1.caption:='немає';
Label48.Caption:=inttostr(Trunc(GetCPUSpeed))+' MHz';
{}
```

Кафедра КБПЗ_2022рік

```
{пам'ять}
memorystatus.dwlength:=sizeof(memorystatus);
globalmemorystatus(memorystatus);
physmemory.caption:=floattostr(memorystatus.dwtotalphys div 1024 div
1024)+' Мега  '+'('+
floattostr(memorystatus.dwtotalphys / 1024 / 1024)+')';
avail.caption:=floattostr(memorystatus.dwavailphys / 1024 / 1024)+' Мег';
maxpf.caption:=floattostr(memorystatus.dwtotalpagefile / 1024 / 1024);
pffree.caption:=floattostr(memorystatus.dwavailpagefile / 1024 / 1024);
{}
{Windows інформація}
winid.caption:=getwinid;
winkey.caption:=getwinkey;
ver1.Caption:=getwinname;
username.caption:=getusernme;
//plusver.caption:=getplusvernum;
company.caption:=getorgname;
resolution.caption:=getscreenresolution;
{printer}
try
getprofilestring('windows','device',',,,',buffer,256);
s:=strpas(buffer);
defprn.Lines.add(' Принтер: '+copy(s,1,pos(',',s)-1));
delete(s,1,pos(',',s)-1);
defprn.lines.add(' Порт: '+copy(s,1,pos(',',s)-1));
delete(s,1,pos(',',s)-1);
defprn.lines.add(' Драйвер і порт:'+ s);
except
showmessage('Printer not found');
end;
{keyboard}
ktype:=GetKeyboardType(0);
case ktype of
     1:keytype.caption:='IBM PC/XT або сумісна ( 83-клавіші)';
     2:keytype.caption:='Olivetti "ICO" ( 102-клавіші)';
     3:keytype.caption:='IBM PC/AT ( 84-клавіші) і інші';
     4:keytype.caption:=' IBM-Розширена (101/ 102-клавіші)';
     5:keytype.caption:='Nokia 1050 та аналогічна клавіатура';
     6:keytype.caption:='Nokia 9140 та аналогічна клавіатура ';
     7:keytype.caption:='Японська клавіатура';
end;
numoffunckey.Caption:=inttostr(getkeyboardtype(2));
{
typ.hide;
label14.hide;
{windir}
getwindowsdirectory(sp,max_path);
wd:=strpas(sp);
{windir.caption:=wd;
progrfiles.caption:=getprogramfilesdir;
label13.hide;
label12.hide;
{Windows version}
OSVerInfo.dwOsversioninfosize:=sizeof(osverinfo);
getversionex(osverinfo);
case osverinfo.dwplatformid of
ver_platform_win32s:os.caption:='Windows 3.x';
ver_platform_win32_windows:os.Caption:='Windows 95 (98)';
ver_platform_win32_nt:os.caption:='Windows NT';
end;
with osverinfo do
begin
winver:=format('%d. %d',[dwmajorversion, dwminorversion]);
build:=format('%d', [LoWord(dwbuildnumber)]);
osver.caption:=winver;
osver.caption:=osver.caption+'  (складання: '+build+')';
end;

{boot}
```

Кафедра_КОПЗ_2022рік

```
{oottype.caption:=getboottype;

{printer}
{Prntrs.items:=Printer.Printers;}
prn.items:=Printer.Printers;
try
fnt.items:=printer.fonts;
except
end;
prn.ItemIndex:=0;
edit2.text:=inttostr(printer.pageheight);
edit1.text:=inttostr(printer.pagewidth);
GetPrName(Processor1);
GetPrName(pt);
resolution.Caption :=inttostr(Screen.Width)+'на'+inttostr(Screen.Height);
timer1.Enabled:=true;
end;

function OpenCD(Drive : Char) : Boolean;
Var
  Res : MciError;
  OpenParm: TMCI_Open_Parms;
  Flags : DWord;
  S : String;
  DeviceID : Word;
begin
  Result := False;
  S := Drive + ':';
  Flags := mci_Open_Type or mci_Open_Element;
  With OpenParm do begin
    dwCallback := 0;
    lpstrDeviceType := 'CDAudio';
    lpstrElementName := PChar(S);
  end;
  {Цей рядок необхідний для правильної роботи функції IntellectCD}
  Res := mciSendCommand(0, mci_Open, Flags, Longint(@OpenParm));
  IF Res <> 0 Then Exit;
  DeviceID := OpenParm.wDeviceID;
  try
    Res:=mciSendCommand(DeviceID, MCI_SET, MCI_SET_DOOR_OPEN, 0);
    IF Res = 0 Then Exit;
    Result := True;
  finally
    mciSendCommand(DeviceID, mci_Close, Flags, Longint(@OpenParm));
  end;
end;
function CloseCD(Drive : Char) : Boolean;
Var
  Res : MciError;
  OpenParm: TMCI_Open_Parms;
  Flags : DWord;
  S : String;
  DeviceID : Word;
begin
  Result := False;
  S := Drive + ':';
  Flags := mci_Open_Type or mci_Open_Element;
  With OpenParm do begin
    dwCallback := 0;
    lpstrDeviceType := 'CDAudio';
    lpstrElementName := PChar(S);
  end;
  Res := mciSendCommand(0, mci_Open, Flags, Longint(@OpenParm));
  IF Res <> 0 Then Exit;
  DeviceID := OpenParm.wDeviceID;
  try
    Res := mciSendCommand(DeviceID, MCI_SET, MCI_SET_DOOR_CLOSED, 0);
    IF Res = 0 Then
    Result := True;
```

Кафедра_КБПЗ_2022рік

```
  finally
    mciSendCommand(DeviceID, mci_Close, Flags, Longint(@OpenParm));
  end;
end;
procedure Delay(msecs : Longint);
var
FirstTick : Longint;
begin
FirstTick := GetTickCount;
repeat
Application.ProcessMessages;
until GetTickCount - FirstTick >= msecs;
end;
procedure TDiadnostic.Button1Click(Sender: TObject);
var disk1:integer;
begin
for disk1:=0 to diskname.items.count-1 do
begin
if CheckDriveType(diskname.items[disk1][1])=' CD-ROM'
then
begin
opencd(diskname.items[disk1][1]);
delay(5000);
closecd(diskname.items[disk1][1]);

end;
end;

end;

procedure TDiadnostic.SpeedButton1Click(Sender: TObject);
begin
form1.show;
end;

procedure TDiadnostic.SpeedButton2Click(Sender: TObject);
begin
//ShellExecute(handle,nil,'mem.exe',nil,nil,sw_restore);
MessageDlg('Тестуюча програма завантажена в оперативну
пам'ять',mtInformation,[mbok],0);
end;
end.
```

Кафедра_КбПЗ_2022рік

Модуль тестування процесора

```
unit ProcessorClockCounter;

interface

uses  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs;

type
  TClockPriority=(cpIdle, cpNormal, cpHigh, cpRealTime, cpProcessDefined);

  TPrecizeProc = procedure(Sender: TObject) of Object;

  TProcessorClockCounter = class(TComponent)
  private
   FCache:array[0..(1 shl 19) - 1] of byte;  // 512 Kb NOP instructions is
enough to clear cache
   FStarted:DWORD;
   FClockPriority:TClockPriority;
   FProcessHandle:HWND;
   FCurrentProcessPriority: Integer;
   FDesiredProcessPriority: Integer;
   FThreadHandle:HWND;
   FCurrentThreadPriority: Integer;
   FDesiredThreadPriority: Integer;
   FCalibration:int64;
   FPrecizeCalibration:int64;
   FStartValue:int64;
   FStopValue:int64;
   FDeltaValue:int64;
   FPrecizeProc:TPrecizeProc;
   FCounterSupported:boolean;
   procedure PrecizeStart;
   procedure PrecizeStartInCache;
   procedure GetProcInf;
   procedure SetClockPriority(Value: TClockPriority);
   procedure ProcedureWithoutInstruction;
   function  GetClock:Int64; register;
   function GetStarted:Boolean;
  protected
   procedure AdjustPriority; virtual; // внутрішнє використання в
конструкторі, щоб встанавлювати параметри коли класс створений в  RunTime
   function  CheckCounterSupported:boolean;
  public
   constructor Create(AOwner: TComponent); override;
   destructor Destroy; override;
   procedure Calibrate;
   procedure Start;
   procedure Stop;
   procedure EraseCache;
   procedure TestPrecizeProc; virtual;
   procedure TestPrecizeProcInCache; virtual;
   property Counter:int64 read FDeltaValue;    // містить значення часу
виміряного тексту (StopValue - StartValue - Calibration)
   property StartValue:int64 read FStartValue; // значення початку
   property StopValue:int64 read FStopValue;   // значення кінця тесту
   property Started:Boolean read GetStarted;
   property CurrentClock:int64 read GetClock;  // для більш довгих тестів це
могло бути використанням, щоб опублікувати поточний лічильник
   property ClockPriority:TClockPriority read FClockPriority write
SetClockPriority default cpNormal;
   property Calibration:int64 read FCalibration; // це використано, щоб
анулювати саме кодову синхронізацію виконання
   property OnPrecizeProc:TPrecizeProc read FPrecizeProc write
FPrecizeProc; // споживач може визначити це для випробування частини коду в
цьому
   property CounterSupported:boolean read FCounterSupported;
  end;
```

Кафедра_КбПЗ_2022рік

```
procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('ASM Utils', [TProcessorClockCounter]);
end;

constructor TProcessorClockCounter.Create(AOwner: TComponent);
var n:integer;
begin
 inherited create(AOwner);
 FCounterSupported:=CheckCounterSupported;
 for n:=0 to High(FCache)-1 do FCache[n]:=$90; // наповнювач з NOP
інструкціями
 FCache[High(FCache)]:=$C3;                    // остання з RET інструкцій
 FClockPriority:=cpNormal;
 FStarted:=0;
 FDesiredProcessPriority:=NORMAL_PRIORITY_CLASS;
 FDesiredThreadPriority :=THREAD_PRIORITY_NORMAL;
 AdjustPriority;
 Calibrate;
 FStartValue:=0;
 FStopValue:=0;
 FDeltaValue:=0;
end;

destructor TProcessorClockCounter.Destroy;
begin
inherited destroy;
end;

procedure TProcessorClockCounter.GetProcInf;
begin
 FProcessHandle:=GetCurrentProcess;
 FCurrentProcessPriority:=GetPriorityClass(FProcessHandle);
 FThreadHandle:=GetCurrentThread;
 FCurrentThreadPriority:=GetThreadPriority(FThreadHandle);
end;

procedure TProcessorClockCounter.AdjustPriority;
begin
GetProcInf;
 case FDesiredProcessPriority of
  IDLE_PRIORITY_CLASS:     FClockPriority:=cpIdle;
  NORMAL_PRIORITY_CLASS:   FClockPriority:=cpNormal;
  HIGH_PRIORITY_CLASS:     FClockPriority:=cpHigh;
  REALTIME_PRIORITY_CLASS: FClockPriority:=cpRealTime;
 end;
end;

procedure TProcessorClockCounter.SetClockPriority(Value: TClockPriority);
begin
 if Value<>FClockPriority then
  begin
   FClockPriority:=Value;
   case FClockPriority of
    cpIdle:    begin
               FDesiredProcessPriority:=IDLE_PRIORITY_CLASS;
               FDesiredThreadPriority :=THREAD_PRIORITY_IDLE;
               end;
    cpNormal:  begin
               FDesiredProcessPriority:=NORMAL_PRIORITY_CLASS;
               FDesiredThreadPriority :=THREAD_PRIORITY_NORMAL;
               end;
    cpHigh:    begin
               FDesiredProcessPriority:=HIGH_PRIORITY_CLASS;
```

Кафедра_КбПЗ_2022рік

```
                FDesiredThreadPriority :=THREAD_PRIORITY_HIGHEST;
                end;
    cpRealTime:begin
                FDesiredProcessPriority:=REALTIME_PRIORITY_CLASS;
                FDesiredThreadPriority :=THREAD_PRIORITY_TIME_CRITICAL;
                end;
    cpProcessDefined:
                begin
                FDesiredProcessPriority:=FCurrentProcessPriority;
                FDesiredThreadPriority :=FCurrentThreadPriority;
                end;
   end;
   Calibrate;
  end;
end;

procedure TProcessorClockCounter.TestPrecizeProc;
begin
FDeltaValue:=0;
if FCounterSupported and assigned(FPrecizeProc) then
 begin
 PrecizeStart;                      // початок тесту
 end;
end;

procedure TProcessorClockCounter.TestPrecizeProcInCache;
begin
FDeltaValue:=0;
if FCounterSupported and  assigned(FPrecizeProc) then
 begin
 EraseCache;
 PrecizeStartInCache;               // перший тест з наповнювачем кешу
процесора

 PrecizeStartInCache;               // другий тест
                                    // згенеруйте величину калібрування
forcode уже вклала кеш
 end;
end;

procedure TProcessorClockCounter.ProcedureWithoutInstruction;
// це використано для калібрування! НЕ ЗМІНЮЙТЕ
asm
 ret
end;

procedure TProcessorClockCounter.EraseCache; register;
asm
  push ebx
  lea ebx,[eax + FCache]
  call ebx                  // примусовий виклик, щоб кодуватися в масиві :)
  pop ebx                   // це заповнить level2 кеш NOPs (Для материнки з 1
Mb рівнями 2 кеша,
  ret                       // розмір масиву повинен бути підвищений на 1 Mb)

// наступні інструкції ніколи не виконані але мали потребу у відповідному
вирівнюватися 16 байтів.
 // Деякі процесори мають інший час виконання коли код - не 16 байтів вирівняних
// Дійсно, наповнювач (у деяких процесорах), внутрішній механізм рівня 1 кеш
(кеш будував
// у процесорі) призначений ловити блоку пам'яті швидше, коли // це - 16 байтів
вирівняних !!!
  nop
  nop
  nop
  nop
  nop
  nop
end;
```

Кафедра_КбПЗ_2022рік

```
function TProcessorClockCounter.GetClock: Int64; register;
asm
 push edx
 push ebx
 push eax
 mov ebx,eax
 xor eax,eax                                // EAX & EDX ініціалізовані в нуль
 mov edx,eax                                // тестує зустрічну підтримку
 DW $310f                                   // Ця інструкція зробить виключення
 sub eax,dword ptr [ebx+FStartValue]    // або зробіть ніщо в процесорах
wthout
 sbb edx,dword ptr [ebx+FStartValue+4]  // зустрічна підтримка
 sub eax,dword ptr [ebx+FCalibration]
 sbb edx,dword ptr [ebx+FCalibration+4]
 mov dword ptr [esp + $10],eax
 mov dword ptr [esp + $14],edx
 pop eax
 pop ebx
 pop edx
 ret
end;

procedure TProcessorClockCounter.PrecizeStartInCache; register;
asm
//this address should be 16 byte aligned
 push edx
 push ebx
 push eax
 mov ebx,eax
 push eax
 mov dword ptr [ebx + FStarted],1           // started:=true

 DW $310f                                   //початок
 mov dword ptr [ebx + FStartValue],eax      // startvalue:=counter
 mov dword ptr [ebx + FStartValue + 4],edx
 mov edx,[ebx + FPrecizeProc + 4]           //дорівнює часу
 mov ebx,ebx
 nop
 nop
 nop
 call ProcedureWithoutInstruction           // процедура виклику з
невідкладним
 DW $310f                                   //СТОП
 mov dword ptr [ebx + FStopValue],eax       // stopvalue:=counter
 mov dword ptr [ebx + FStopValue + 4],edx
 sub eax,dword ptr [ebx + FStartValue]
 sbb edx,dword ptr [ebx + FStartValue + 4]
 mov dword ptr [ebx + FPrecizeCalibration],eax     //
calibration:=stopvalue - startvalue
 mov dword ptr [ebx + FPrecizeCalibration + 4],edx
 nop                                         // необхідно відповідному
виравнюванні
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 DW $310f                                   //початок
 mov dword ptr [ebx + FStartValue],eax      // startvalue:=counter
 mov dword ptr [ebx + FStartValue + 4],edx
 mov eax,[ebx + FPrecizeProc + 4]
```

Кафедра_КбПЗ_2022рік

```
 mov edx,ebx
 call [ebx + FPrecizeProc]
 DW $310f                                    //СТОП
 pop ebx
 mov dword ptr [ebx + FStopValue],eax        // stopvalue:=counter
 mov dword ptr [ebx + FStopValue + 4],edx
 sub eax,dword ptr [ebx + FStartValue]
 sbb edx,dword ptr [ebx + FStartValue + 4]
 sub eax,dword ptr [ebx + FPrecizeCalibration]
 sbb edx,dword ptr [ebx + FPrecizeCalibration + 4]
 mov dword ptr [ebx + FDeltaValue],eax       // deltavalue:=stopvalue -
startvalue - calibration
 mov dword ptr [ebx + FDeltaValue + 4],edx
 pop eax
 pop ebx
 pop edx
 ret
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
end;

procedure TProcessorClockCounter.PrecizeStart; register;
asm
// ця адреса повинен бути 16 байтів вирівняних
 push edx
 push ebx
 push eax
 call EraseCache                             // кеш наповнювач з NOPs
виконуючи це
 mov ebx,eax
 push eax
 mov dword ptr [ebx + FStarted],1            // почате:=true
 nop                                         // потреба у відповідному
вирівнюється
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 nop
 DW $310f                                    //початок
 mov dword ptr [ebx + FStartValue],eax       // startvalue:=counter
 mov dword ptr [ebx + FStartValue + 4],edx
 mov edx,[ebx + FPrecizeProc + 4]            //дорівнюється часу
 mov ebx,ebx
 nop
 nop
 nop
 call ProcedureWithoutInstruction            // процедура виклика з
невідкладним
 DW $310f                                    //СТОП
 mov dword ptr [ebx + FStopValue],eax        // stopvalue:=counter
 mov dword ptr [ebx + FStopValue + 4],edx
 sub eax,dword ptr [ebx + FStartValue]
 sbb edx,dword ptr [ebx + FStartValue + 4]
```

Кафедра_КбПЗ_2022рік

```
 mov dword ptr [ebx + FPrecizeCalibration],eax     //
calibration:=stopvalue - startvalue
 mov dword ptr [ebx + FPrecizeCalibration + 4],edx
 mov eax,ebx
 call EraseCache;                                   // кеш наповнювач с NOPs
виконує це
 nop                                                // потреба у відповідному
виравнює
 nop
 nop
 nop
 nop
 DW $310f                                          //початок
 mov dword ptr [ebx + FStartValue],eax             // startvalue:=counter
 mov dword ptr [ebx + FStartValue + 4],edx
 mov eax,[ebx + FPrecizeProc + 4]
 mov edx,ebx
 call [ebx + FPrecizeProc]
 DW $310f                                          //СТОП
 pop ebx
 mov dword ptr [ebx + FStopValue],eax              // stopvalue:=counter
 mov dword ptr [ebx + FStopValue + 4],edx
 sub eax,dword ptr [ebx + FStartValue]
 sbb edx,dword ptr [ebx + FStartValue + 4]
 sub eax,dword ptr [ebx + FPrecizeCalibration]
 sbb edx,dword ptr [ebx + FPrecizeCalibration + 4]
 mov dword ptr [ebx + FDeltaValue],eax             // deltavalue:=stopvalue -
startvalue - calibration
 mov dword ptr [ebx + FDeltaValue + 4],edx
 pop eax
 pop ebx
 pop edx
end;


end.
```

Кафедра_КбПЗ_2022рік

Модуль діагностики

```
unit Systeminfo;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,extctrls;

type TDialupAdapterInfo = record //Інформація про Dialup адаптер
  alignment:dword;
  buffer:dword;
  bytesrecieved:dword;
  bytesXmit:dword;
  ConnectSpeed:dword;
  CRC:dword;
  framesrecieved:dword;
  FramesXmit:dword;
  Framing:dword;
  runts:dword;
  Overrun:dword;
  timeout:dword;
  totalbytesrecieved:dword;
  totalbytesXmit:dword;
end;

type TKernelInfo = record
  CpuUsagePcnt:dword;
  Numthreads:dword;
  NumVMS:dword;
end;

type TFATInfo = record
  BreadsSec:dword;
  BwritesSec:dword;
  Dirtydata:dword;
  ReadsSec:dword;
  WritesSec:dword;
end;

type TVMMInfo = record
  CDiscards:dword;
  CInstancefaults:dword;
  CPageFaults:dword;
  cPageIns:dword;
  cPageOuts:dword;
  cpgCommit:dword;
  cpgDiskCache:dword;
  cpgDiskCacheMac:dword;
  cpgDiskCacheMid:dword;
  cpgDiskCacheMin:dword;
  cpgfree:dword;

  cpglocked:dword;
  cpglockedNoncache:dword;
  cpgother:dword;
  cpgsharedpages:dword;
  cpgswap:dword;
  cpgswapfile:dword;
  cpgswapfiledefective:dword;
  cpgswapfileinuse:dword;
end;

type
  TSysInfo = class(TComponent)
  private
   fDialupAdapterInfo:TDialupAdapterInfo;
   fKernelInfo:TKernelInfo;
```

Кафедра_КбПЗ_2022рік

```
    fVCACHEInfo:TVCACHEInfo;
    fFATInfo:TFATInfo;
    fVMMInfo:TVMMInfo;
    ftimer:TTimer;
    fupdateinterval:integer;
    tmp:dword;
    vsize:dword;
    pkey:hkey;
    regtype:pdword;
    fstopped:boolean;
    procedure fupdatinginfo(sender:tobject);
    procedure fsetupdateinterval(aupdateinterval:integer);
  protected
    fsysInfoChanged:TNotifyEvent;
  public
    constructor Create(Aowner:Tcomponent);override;
    destructor  Destroy;override;

    property DialupAdapterInfo: TDialupAdapterInfo read fDialupAdapterInfo;
    property KernelInfo: TKernelInfo read fKernelInfo;
    property VCACHEInfo: TVCACHEInfo read fVCACHEInfo;
    property FATInfo: TFATInfo read fFATInfo;
    property VMMInfo: TVMMInfo read fVMMInfo;
    procedure StartRecievingInfo;
    procedure StopRecievingInfo;
  published
    property SysInfoChanged:TNotifyEvent read fsysInfoChanged write
      fsysInfoChanged;//Ця подія викликається після певного інтервалу часу.
    property UpdateInterval:integer read fupdateInterval write
      fsetupdateinterval default 5000;
  end;

procedure TSysInfo.startrecievingInfo;
var
res:integer;
begin
res:=RegOpenKeyEx(HKEY_DYN_DATA,'PerfStats\StartStat',0,KEY_ALL_ACCESS,pke
y);
if res<>0 then
  raise exception.Create('Не можу івдкрити ключ реєстрації');
fstopped:=false;
// Для Dial Up Адаптера
RegQueryValueEx(pkey,' Dial-Up
Adapter\Alignment',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up Adapter\Buffer',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up Adapter\Framing',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up Adapter\Overrun ',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up Adapter\Timeout',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up Adapter\CRC',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up Adapter\Runts',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\FramesXmit',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\FramesRecvd',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\BytesXmit',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\BytesRecvd',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\TotalBytesXmit',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\TotalBytesRecvd',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\ConnectSpeed',nil,regtype,@tmp,@vsize);

// Для VCACHE
RegQueryValueEx(pkey,'VCACHE\LRUBuffers',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VCACHE\FailedRecycles',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VCACHE\RandomRecycles',nil,regtype,@tmp,@vsize);
```

Кафедра_КбПЗ_2022рік

```
RegQueryValueEx(pkey,'VCACHE\LRURecycles',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VCACHE\Misses',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VCACHE\Hits',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VCACHE\cMacPages',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VCACHE\cMinPages',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VCACHE\cCurPages',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,' Dial-Up
Adapter\BytesXmit',nil,regtype,@tmp,@vsize);

//Для VFAT

RegQueryValueEx(pkey,'VFAT\DirtyData',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VFAT\BReadsSec',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VFAT\BWritesSec',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VFAT\ReadsSec',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VFAT\WritesSec',nil,regtype,@tmp,@vsize);
//Для VMM

RegQueryValueEx(pkey,'VMM\cpgLockedNoncache',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgCommit',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgSharedPages',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgDiskcacheMid',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgDiskcacheMac',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgDiskcacheMin',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgDiskcache',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgSwapfileDefective',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgSwapfileInUse',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgSwapfile',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cDiscards',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cPageOuts',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cPageIns',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cInstanceFaults',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cPageFaults',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgOther',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgSwap',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgLocked',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'VMM\cpgFree',nil,regtype,@tmp,@vsize);
//Для KERNEL
RegQueryValueEx(pkey,'KERNEL\CPUUsage',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'KERNEL\VMs',nil,regtype,@tmp,@vsize);
RegQueryValueEx(pkey,'KERNEL\Threads',nil,regtype,@tmp,@vsize);
RegCloseKey(pkey);
ftimer.enabled:=true;
end;
destructor tsysinfo.Destroy;
begin
StopRecievingInfo;
ftimer.Destroy;
inherited;
end;
procedure Register;
begin
  RegisterComponents('Samples', [TSysInfo]);
end;
end.
```

Кафедра_КСПС_2012рік

Модуль діагностики процесора

```
unit CPU;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, ProcessorClockCounter, StdCtrls;

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    StaticText1: TStaticText;
    Button7: TButton;
    Button8: TButton;
    procedure pcc1PrecizeProc(Sender: TObject);
    procedure pcc2PrecizeProc(Sender: TObject);
    procedure pcc3PrecizeProc(Sender: TObject);
    procedure pcc4PrecizeProc(Sender: TObject);
    procedure pcc5PrecizeProc(Sender: TObject);
    procedure pcc7PrecizeProc(Sender: TObject);
    procedure pcc8PrecizeProc(Sender: TObject);
procedure Button7Click(Sender: TObject);
    procedure Button8Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

// Тактова частота

procedure TForm1.pcc1PrecizeProc(Sender: TObject);
begin
sleep(1000);  //чекає 1 сек
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
button1.Caption:='Зачекайте';
button1.Enabled:=false;
 pcc1.TestPrecizeProcInCache;
 label1.Caption:=IntToStr(pcc1.Counter)+' Hz';
button1.Caption:='Виміряти тактову частоту';
button1.Enabled:=true;
end;

// швидкість виконання арифметичних операцій

procedure TForm1.pcc2PrecizeProc(Sender: TObject);
var n:integer;
    m:integer;                // ціла змінна
begin
 for n:=0 to 99 do m:=m+1;
end;

procedure TForm1.pcc3PrecizeProc(Sender: TObject);
var n:integer;
    m:Int64;                  // Int64 змінна
begin
```

Кафедра_КбПЗ_2022рік

```
 for n:=0 to 99 do m:=m+1;
end;

procedure TForm1.pcc4PrecizeProc(Sender: TObject);
var n:integer;
    m:single;                  // single type змінна
begin
 for n:=0 to 99 do m:=m + 1.0001;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
 pcc2.TestPrecizeProcInCache;
 label2.Caption:=IntToStr(pcc2.Counter)+' тактів';
 pcc3.TestPrecizeProcInCache;
 label3.Caption:=IntToStr(pcc3.Counter)+' тактів';
 pcc4.TestPrecizeProcInCache;
 label4.Caption:=IntToStr(pcc4.Counter)+' тактів';
end;

// швидкість системної шини

procedure TForm1.pcc5PrecizeProc(Sender: TObject);
begin
 asm
 nop; nop; nop; nop; nop; nop; nop; nop;
 nop; nop; nop; nop; nop; nop; nop; nop;
 nop; nop; nop; nop; nop; nop; nop; nop;
 nop; nop; nop; nop; nop; nop; nop; nop;
 nop; nop; nop; nop; nop; nop; nop; nop;
 nop; nop; nop; nop; nop; nop; nop; nop;
 nop; nop; nop; nop; nop; nop; nop; nop;
 nop; nop; nop; nop; nop; nop; nop; ret;
 end;
end;

procedure TForm1.Button3Click(Sender: TObject);
var cInRAM, cInCache:int64;
begin
 pcc5.TestPrecizeProc;        // Код в RAM та буде владений в КЕШ
 cInRAM:=pcc5.Counter;
 label5.Caption:=IntToStr(cInRAM)+' тактів';
 pcc5.TestPrecizeProcInCache; // код вже у кеші
 cInCache:=pcc5.Counter;
 label6.Caption:=IntToStr(cInCache)+' тактів';
 label7.Caption:=IntToStr( cInRAM-cInCache)+ ' тактів';
end;

// швидкість виклику додатків

procedure TForm1.Button4Click(Sender: TObject);
begin

 pcc6.Start;
 WinExec(PChar('Notepad.exe'),SW_SHOWNORMAL);
 pcc6.Stop;

label8.Caption:=IntToStr(pcc6.Counter)+' тактів';
end;

// Example 5

procedure TForm1.pcc7PrecizeProc(Sender: TObject);
begin
refresh;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
```

Кафедра КбПЗ_2022рік

```
 pcc7.TestPrecizeProcInCache;
 label9.Caption:=IntToStr(pcc7.Counter)+ ' тактів';
end;

// швидкість заповнення кешу

procedure TForm1.pcc8PrecizeProc(Sender: TObject);
begin
asm nop end;
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
 pcc8.TestPrecizeProcInCache;
 label10.Caption:=IntToStr(pcc8.Counter)+ ' тактів';
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
MessageDlg('NOP - Порожня операция'#13 +
'це псевдонім інструкції XCHG (E)AX, (E)AX',
mtInformation,[mbok],0);
end;
procedure TForm1.Button8Click(Sender: TObject);
begin
MessageDlg('процесор Pentium IV'#13 +
'із частотою системної шини 400 Мгц',
mtInformation,[mbok],0);
end;
end.
```

Кафедра_КбПЗ_2022рік