

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН
MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE REPUBLIC OF KAZAKHSTAN



**ХАЛЫҚАРАЛЫҚ АҚПАРАТТЫҚ ЖӘНЕ
КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР
ЖУРНАЛЫ**

**МЕЖДУНАРОДНЫЙ ЖУРНАЛ
ИНФОРМАЦИОННЫХ И
КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ**

**INTERNATIONAL JOURNAL OF INFORMATION
AND COMMUNICATION TECHNOLOGIES**

2024 (17) 1
Қаңтар – наурыз

ISSN 2708–2032 (print)
ISSN 2708–2040 (online)

БАС РЕДАКТОР:

Хикметов Аскар Кусупбекович — басқарма төрағасы, Халықаралық ақпараттық технологиялар университетінің ректоры, физика-математика ғылымдарының кандидаты (Қазақстан)

БАС РЕДАКТОРДЫҢ ОРЫНБАСАРЫ:

Колесникова Катерина Викторовна — техника ғылымдарының докторы, Халықаралық ақпараттық технологиялар университеті, «Ақпараттық жүйелер» кафедрасының проректоры (Қазақстан)

ҒАЛЫМ ХАТШЫ:

Ипалакова Мадина Тулегеновна — техника ғылымдарының кандидаты, қауымдастырылған профессор, «Халықаралық ақпараттық технологиялар университеті» АҚ, Ғылыми-зерттеу жұмыс департаментінің директоры (Қазақстан)

РЕДАКЦИЯЛЫҚ АЛҚА:

Разак Абдул — PhD, Халықаралық ақпараттық технологиялар университетінің профессоры (Қазақстан)

Лучио Томмазо де Паолис — Саленто университетінің (Италия) инновациялар және технологиялық инженерия департаменті AVR зертханасының зерттеу және әзірлеу бөлімінің директоры

Лиз Бэкон — профессор, Абертей университеті вице-канцлердің орынбасары (Ұлыбритания)

Микеле Пагано — PhD, Пиза университетінің профессоры (Италия)

Отелбаев Мухтарбай Отелбаевич — физика-математика ғылымдарының докторы, ҚР ҰҒА академигі, Халықаралық ақпараттық технологиялар университеті, «Математикалық және компьютерлік модельдеу» кафедрасының профессоры (Қазақстан)

Рысбайұлы Болатбек — физика-математика ғылымдарының докторы, Халықаралық ақпараттық технологиялар университеті, «Математикалық және компьютерлік модельдеу» кафедрасының профессоры (Қазақстан)

Дайнеко Евгения Александровна — PhD, қауымдастырылған профессор, Халықаралық ақпараттық технологиялар университетінің Жабандық серіктестік және қосымша білім беру жөніндегі проректоры (Қазақстан)

Дузбаев Нуржан Токсужаевич — PhD, Халықаралық ақпараттық технологиялар университетінің Цифрландыру және инновациялар жөніндегі проректоры (Қазақстан)

Синчев Бахтгерей Куспанович — техника ғылымдарының докторы, Халықаралық ақпараттық технологиялар университетінің «Ақпараттық жүйелер» кафедрасының профессоры (Қазақстан)

Сейлова Нүргүл Абдуллаевна — техника ғылымдарының кандидаты, Халықаралық ақпараттық технологиялар университетінің «Компьютерлік технологиялар және киберқауіпсіздік» факультетінің деканы (Қазақстан)

Мухамедиева Ардак Габитовна — экономика ғылымдарының кандидаты, Халықаралық ақпараттық технологиялар университетінің «Цифрлық трансформациялар» факультетінің деканы (Қазақстан)

Ыдырыс Айжан Жұмабайқызы — PhD, Халықаралық ақпараттық технологиялар университетінің «Математикалық және компьютерлік модельдеу» кафедрасының менгерушісі (Қазақстан)

Шильдибеков Ерлан Жаржанович — PhD, Халықаралық ақпараттық технологиялар университетінің «Экономика және бизнес» кафедрасының менгерушісі (Қазақстан)

Аманжолова Сауле Токсановна — техника ғылымдарының кандидаты, Халықаралық ақпараттық технологиялар университетінің «Киберқауіпсіздік» кафедрасының менгерушісі (Қазақстан)

Ниязгулова Айгүл Аскарбековна — филология ғылымдарының кандидаты, Халықаралық ақпараттық технологиялар университетінің «Медиакоммуникациялар және Қазақстан тарихы» кафедрасының менгерушісі (Қазақстан)

Айтмағамбетов Алтай Зуфарович — техника ғылымдарының кандидаты, Халықаралық ақпараттық технологиялар университетінің «Радиотехника, электроника және телекоммуникация» кафедрасының профессоры (Қазақстан)

Алмисреб Али Абд — PhD, Халықаралық ақпараттық технологиялар университетінің қауымдастырылған профессоры (Қазақстан)

Мохамед Ахмед Хамада — PhD, Халықаралық ақпараттық технологиялар университетінің «Ақпараттық жүйелер» кафедрасының қауымдастырылған профессоры (Қазақстан)

Янг Им Чу — PhD, Гачон университетінің профессоры (Оңтүстік Корея)

Тадеуш Валлас — PhD, Адам Мицкевич атындағы университеттің проректоры (Польша)

Мамырбаев Өркен Жұмажанұлы — Ақпараттық жүйелер саласындағы техника ғылымдарының (PhD) докторы, ҚР БҒМ ҚҰО ақпараттық және есептеу технологиялары институты директорының ғылым жөніндегі орынбасары (Қазақстан)

Бушуев Сергей Дмитриевич — техника ғылымдарының докторы, профессор, Украинаның «УКРНЕТ» жобаларды басқару қауымдастығының директоры, Киев ұлттық құрылыс және сәулет университетінің «Жобаларды басқару» кафедрасының менгерушісі (Украина)

Белолицкая Светлана Васильевна — техника ғылымдарының докторы, доцент, Астана IT университетінің деректер жөніндегі есептеу және ғылым кафедрасының профессоры (Қазақстан)

ЖАУАПТЫ РЕДАКТОР:

Мрзабаева Раушан Жалиевна — «Халықаралық ақпараттық технологиялар университеті» АҚ (Қазақстан)

Халықаралық ақпараттық және коммуникациялық технологиялар журналы

ISSN 2708–2032 (print)

ISSN 2708–2040 (online)

Меншіктенуші: «Халықаралық ақпараттық технологиялар университеті» АҚ (Алматы қ.)

Қазақстан Республикасы Ақпарат және әлеуметтік даму министрлігінің Ақпарат комитетінде – 20.02.2020 жылы берілген.

№ KZ82VPY00020475 мерзімдік басылым тіркеуіне қойылу туралы куәлік.

Тақырыптық бағыты: ақпараттық технологиялар, әлеуметтік-экономикалық жүйелерді дамытудағы цифрлық технологиялар, ақпараттық қауіпсіздік және коммуникациялық технологияларға арналған.

Мерзімділігі: жылына 4 рет.

Тиражы: 100 дана

Редакцияның мекенжайы: 050040, Алматы қ-сы, Манас к-сі, 34/1, 709-кабинет, тел: +7 (727) 244-51-09.

E-mail: ijict@iitu.edu.kz

Журнал сайты: <https://journal.iitu.edu.kz>

© Халықаралық ақпараттық технологиялар университеті АҚ, 2024

© Авторлар ұжымы, 2024

ГЛАВНЫЙ РЕДАКТОР:

Хикметов Аскар Кусулбекович — кандидат физико-математических наук, председатель правления - ректор Международного университета информационных технологий (Казахстан)

ЗАМЕСТИТЕЛЬ ГЛАВНОГО РЕДАКТОРА:

Колесникова Катерина Викторовна — доктор технических наук, профессор, проректор по научно-исследовательской деятельности Международного университета информационных технологий (Казахстан)

УЧЕНЫЙ СЕКРЕТАРЬ:

Ипалакова Мадина Тулегеновна — кандидат технических наук, ассоциированный профессор, директор департамента по научно-исследовательской деятельности Международного университета информационных технологий (Казахстан)

РЕДАКЦИОННАЯ КОЛЛЕГИЯ:

Разак Абдул — PhD, профессор кафедры кибербезопасности Международного университета информационных технологий (Казахстан)

Лучно Томмазо де Паолис — директор отдела исследований и разработок лаборатории AVR департамента инноваций и технологического инжиниринга Университета Саленто (Италия)

Лиз Бэкон — профессор, заместитель вице-канцлера Университета Абертей (Великобритания)

Микеле Пагано — PhD, профессор Университета Пизы (Италия)

Отелбаев Мухтарбай Отелбайулы — доктор физико-математических наук, профессор, академик НАН РК, профессор кафедры математического и компьютерного моделирования Международного университета информационных технологий (Казахстан)

Рысбайулы Болатбек — доктор физико-математических наук, профессор, профессор кафедры математического и компьютерного моделирования Международного университета информационных технологий (Казахстан)

Дайнеко Евгения Александровна — PhD, ассоциированный профессор, проректор по глобальному партнерству и дополнительному образованию Международного университета информационных технологий (Казахстан)

Дузбаев Нуржан Токкужаевич — PhD, ассоциированный профессор, проректор по цифровизации и инновациям Международного университета информационных технологий (Казахстан)

Синчев Бахтгерей Куспанович — доктор технических наук, профессор, профессор кафедры информационных систем Международного университета информационных технологий (Казахстан)

Сейлова Нургуль Абадуллаевна — кандидат технических наук, декан факультета компьютерных технологий и кибербезопасности Международного университета информационных технологий (Казахстан)

Мухамедиева Ардак Габитовна — кандидат экономических наук, декан факультета цифровых трансформаций Международного университета информационных технологий (Казахстан)

Ыдырыс Айжан Жумабаевна — PhD, ассистент профессор, заведующая кафедрой математического и компьютерного моделирования Международного университета информационных технологий (Казахстан)

Шилдибеков Ерлан Жаржанович — PhD, заведующий кафедрой экономики и бизнеса Международного университета информационных технологий (Казахстан)

Аманжолова Сауле Токсановна — кандидат технических наук, заведующая кафедрой кибербезопасности Международного университета информационных технологий (Казахстан)

Ниязгулова Айгуль Аскарбековна — кандидат филологических наук, доцент, заведующая кафедрой медиакоммуникаций и истории Казахстана Международного университета информационных технологий (Казахстан)

Айтмагамбетов Алтай Zufарович — кандидат технических наук, профессор кафедры радиотехники, электроники и телекоммуникаций Международного университета информационных технологий (Казахстан)

Алмисреб Али Абд — PhD, ассоциированный профессор кафедры кибербезопасности Международного университета информационных технологий (Казахстан)

Мохамед Ахмед Хамада — PhD, ассоциированный профессор кафедры информационных систем Международного университета информационных технологий (Казахстан)

Янг Им Чу — PhD, профессор университета Гачон (Южная Корея)

Тадеш Валлас — PhD, проректор университета имен Адама Мицкевича (Польша)

Мамырбаев Оркен Жумажанович — PhD, заместитель директора по науке РГП Института информационных и вычислительных технологий Комитета науки МНВО РК (Казахстан)

Бушуев Сергей Дмитриевич — доктор технических наук, профессор, директор Украинской ассоциации управления проектами «УКРНЕТ», заведующий кафедрой управления проектами Киевского национального университета строительства и архитектуры (Украина)

Белошницкая Светлана Васильевна — доктор технических наук, доцент, профессор кафедры вычислений и науки о данных Astana IT University (Казахстан)

ОТВЕТСТВЕННЫЙ РЕДАКТОР:

Мрзабаева Раушан Жалиевна — АО «Международный университет информационных технологий» (Казахстан).

Международный журнал информационных и коммуникационных технологий

ISSN 2708–2032 (print)

ISSN 2708–2040 (online)

Собственник: АО «Международный университет информационных технологий» (г. Алматы).

Свидетельство о постановке на учет периодического печатного издания в Министерство информации и общественного развития Республики Казахстан № KZ82VPY00020475, выданное от 20.02.2020 г.

Тематическая направленность: информационные технологии, информационная безопасность и коммуникационные технологии, цифровые технологии в развитии социо-экономических систем.

Периодичность: 4 раза в год.

Тираж: 100 экземпляров.

Адрес редакции: 050040 г. Алматы, ул. Манаса 34/1, каб. 709, тел: +7 (727) 244-51-09.

E-mail: ijict@iitu.edu.kz

Сайт журнала: <https://journal.iitu.edu.kz>

© АО Международный университет информационных технологий, 2024

© Коллектив авторов, 2024

EDITOR-IN-CHIEF:

Khikmetov Askar Kusupbekovich — Candidate of Physical and Mathematical Sciences, Chairman of the Board, Rector of International Information Technology University (Kazakhstan)

DEPUTY CHIEF DIRECTOR:

Kolesnikova Katerina Viktorovna — Doctor of Technical Sciences, Vice-Rector of Information Systems Department, International Information Technology University (Kazakhstan)

SCIENTIFIC SECRETARY:

Ipalakova Madina Tulegenovna — Candidate of Technical Sciences, Associate Professor, Director of the Research Department, International University of Information Technologies (Kazakhstan)

EDITORIAL BOARD:

Razaq Abdul — PhD, Professor of International Information Technology University (Kazakhstan)

Lucio Tommaso de Paolis — Director of Research and Development, AVR Laboratory, Department of Innovation and Process Engineering, University of Salento (Italy)

Liz Bacon — Professor, Deputy Director, and Deputy Vice-Chancellor of the University of Abertay. (Great Britain)

Michele Pagano — Ph.D., Professor, University of Pisa (Italy)

Otelbaev Mukhtarbay Otelbayuly — Doctor of Physical and Mathematical Sciences, Academician of the National Academy of Sciences of the Republic of Kazakhstan, Professor of the Department of Mathematical and Computer Modeling of International Information Technology University (Kazakhstan)

Rysbayuly Bolatbek — Doctor of Physical and Mathematical Sciences, Professor of the Department of Mathematical and Computer Modeling, International Information Technology University (Kazakhstan)

Daineko Yevgeniya Alexandrovna — PhD, Associate Professor, Vice-Rector for Global Partnership and Continuing Education, International Information Technology University (Kazakhstan)

Duzbaev Nurzhan Tokkuzhaevich — Candidate of Technical Sciences, Vice-Rector for Digitalization and Innovations, International Information Technology University (Kazakhstan)

Sinchev Bakhtgeray Kuspanuly — Doctor of Technical Sciences, Professor of the Department of Information Systems, International Information Technology University (Kazakhstan)

Seilova Nurgul Abdullaevna — Candidate of Technical Sciences, Dean of the Faculty of Computer Technologies and Cybersecurity, International Information Technology University (Kazakhstan)

Mukhamedieva Ardak Gabitovna — Candidate of Economic Sciences, Dean of the Faculty of Digital Transformations, International Information Technology University (Kazakhstan)

Idyrys Aizhan Zhumabaevna — PhD, Head of the Department of Mathematical and Computer Modeling, International Information Technology University (Kazakhstan)

Shildibekov Yerlan Zharzhanuly — PhD, Head of the Department of Economics and Business, International Information Technology University (Kazakhstan)

Amanzholova Saule Toksanovna — Candidate of Technical Sciences, Head of the Department of Cyber Security, International Information Technology University (Kazakhstan)

Niyazgulova Aigul Askarbekovna — Candidate of Philology, Head of the Department of Media Communications and History of Kazakhstan, International Information Technology University (Kazakhstan)

Aitmagambetov Altai Zufarovich — Candidate of Technical Sciences, Professor of the Department of Radioengineering, Electronics and Telecommunication, International Information Technology University (Kazakhstan)

Almisreb Ali Abd — PhD, Associate Professor, International Information Technology University (Kazakhstan)

Mohamed Ahmed Hamada — PhD, Associate Professor, Department of Information systems, International Information Technology University (Kazakhstan)

Young Im Choo — PhD, Professor, Gachon University (South Korea)

Tadeusz Wallas — PhD, University of Dr. Litt Adam Miskevich in Poznan (Poland)

Mamyrbayev Orken Zhumazhanovich — PhD in Information Systems, Deputy Director for Science, Institute of Information and Computing Technologies CS MSHE RK (Kazakhstan)

Bushuyev Sergey Dmitriyevich — Doctor of Technical Sciences, Professor, Director of Удoктoр тeхнических наук, профессор, директор Ukrainian Association of Project Management UKRNET, Head of Project Management Department, Kyiv National University of Construction and Architecture (Ukraine)

Beloshitskaya Svetlana Vasilyevna — Doctor of Technical Sciences, Associate Professor, Professor of the Department of Computing and Data Science, Astana IT University (Kazakhstan)

EXECUTIVE EDITOR

Mrzabayeva Raushan Zhalievna — International Information Technology University (Kazakhstan)

«International Journal of Information and Communication Technologies»

ISSN 2708–2032 (print)

ISSN 2708–2040 (online)

Owner: International Information Technology University JSC (Almaty).

The certificate of registration of a periodical printed publication in the Ministry of Information and Social Development of the Republic of Kazakhstan, Information Committee No. KZ82VPY00020475, issued on 20.02.2020.

Thematic focus: information technology, digital technologies in the development of socio-economic systems, information security and communication technologies

Periodicity: 4 times a year.

Circulation: 100 copies.

Editorial address: 050040. Manas st. 34/1, Almaty. +7 (727) 244-51-09. E-mail: ijict@iitu.edu.kz

Journal website: <https://journal.iitu.edu.kz>

© International Information Technology University JSC, 2024

© Group of authors, 2024

МАЗМҰНЫ

ӘЛЕУМЕТТІК-ЭКОНОМИКАЛЫҚ ЖҮЙЕЛЕРДІ ДАМЫТУДАҒЫ ЦИФРЛЫҚ ТЕХНОЛОГИЯЛАР

А. Агдавлетова, В. Мадин, О. Салыкова БАҒДАРЛАМАНАТЫН ЛОГИКАЛЫҚ КОНТРОЛЛЕРДЕ (БЛК) ТЕРЕҢ ОҚЫТУ АРҚЫЛЫ ТЕХНОЛОГИЯЛЫҚ ҮДЕРІСТЕРДІ АДАПТИВТІ БАСҚАРУ.....	8
Ф. Бхат, Н.А. Сейлова, В.В. Покусов КОМПЬЮТЕРЛЕРДІ ЖҮКТЕМЕЛІК ТЕСТІЛЕУ БОЙЫНША СЫНАҚТАР ӘДІСТЕМЕСІН ЖҮРГІЗУ БАҒДАРЛАМАСЫН ЖҮЗЕГЕ АСЫРУ.....	29
Ж.М. Досхожина ҚАЗІРГІ ӘЛЕМДЕГІ МӘДЕНИЕТАРАЛЫҚ КОММУНИКАЦИЯ ПРИНЦИПТЕР.....	48
Ұ.Р. Ералиев ӘСКЕРИ САЛА ХАЛЫҚАРАЛЫҚ ҚАТЫНАСТАРДЫҢ ҚҰРАМДАС БӨЛГІ РЕТІНДЕ.....	56
А. Төлеубеков, А. Досқожанова ҚАЗІРГІ ТЕХНОЛОГИЯЛАР: ХАЙДЕГГЕРДІҢ ТӘСІЛДЕРІ.....	63

АҚПАРАТТЫҚ ТЕХНОЛОГИЯЛАР

А. Маратұлы, Е.А. Абибуллаев YOLO-NAS ЖӘНЕ YOLO-НЫҢ АЛДЫҢҒЫ НҮСҚАЛАРЫНЫҢ ӨНІМДІЛІГІН ЗЕРТТЕУ ЖӘНЕ САЛЫСТЫРМАЛЫ ТАЛДАУ.....	71
Е.Е. Мұратханов, Е.А. Жанбабаев ХАЛЫҚАРАЛЫҚ ҚАТЫНАС ОРНАТУ КЕЗІНДЕГІ IT-ТЕХНОЛОГИЯЛАРЫНЫҢ ҚАЖЕТТІЛІГІ.....	84
К.М. Шертаев, Л. Ниязбаева СПИКЕРДІ АНЫҚТАУДА ТЕРЕҢ ОҚУ: ЗАМАНАУ ӘДІСТЕР ЖӘНЕ ДАМУ БОЛАШАҒЫ.....	98

АҚПАРАТТЫҚ ҚАУІПСІЗДІК ЖӘНЕ КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАРҒА АРНАЛҒАН

Д. Лукьянов, А. Колесников ISV 4.0 IPMA МЫСАЛЫ БОЙЫНША ЖОБАНЫ БАСҚАРУ САЛАСЫНДАҒЫ БІЛІМ ЖҮЙЕЛЕРІН ТАЛДАУДА ЭНТРОПИЯ ТӘСІЛІ ПАЙДАЛАНУ	110
П.С. Пустовойтов, Н.А. Сейлова, А.С. Гнатюк ДАУЫС ЖАЛҒАН ӘДІСТЕРІН ТАЛДАУ: ТӘУЕКЕЛДЕР, ЖАҒДАЙЛАР ЖӘНЕ ҚОРҒАУ СТРАТЕГИЯЛАРЫ.....	122

СОДЕРЖАНИЕ

ЦИФРОВЫЕ ТЕХНОЛОГИИ В РАЗВИТИИ СОЦИО-ЭКОНОМИЧЕСКИХ СИСТЕМ

А. Агдавлетова, В. Мадин, О. Салыкова АДАПТИВНОЕ УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ С ПОМОЩЬЮ ГЛУБОКОГО ОБУЧЕНИЯ НА ПРОГРАММИРУЕМОМ ЛОГИЧЕСКОМ КОНТРОЛЛЕРЕ (ПЛК).....	8
Ф. Бхат, Н.А. Сейлова, В.В. Покусов ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДИКИ ПРОВЕДЕНИЯ ИСПЫТАНИЙ И НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ КОМПЬЮТЕРОВ.....	29
Ж.М. Досхожина ПРИНЦИПЫ МЕЖКУЛЬТУРНОЙ КОММУНИКАЦИИ В СОВРЕМЕННОМ МИРЕ.....	48
У.Р. Ералиев ВОЕННАЯ СФЕРА КАК СОСТАВЛЯЮЩАЯ МЕЖДУНАРОДНЫХ ОТНОШЕНИЙ.....	56
А. Тулеубеков, А. Доскожанова О СОВРЕМЕННЫХ ТЕХНОЛОГИЯХ: ПОДХОД ХАЙДЕГГЕРА.....	63

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

А. Маратулы, Е.А. Абибуллаев ИССЛЕДОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ YOLO-NAS И ПРЕДЫДУЩИХ ВЕРСИЙ YOLO.....	71
Е.Е. Муратханов, Е.А. Жанбабаев ВАЖНОСТЬ IT-ТЕХНОЛОГИЙ В УСТАНОВЛЕНИИ МЕЖДУНАРОДНЫХ ОТНОШЕНИЙ.....	84
К.М. Шертаев, Л. Ниязбаева ГЛУБОКОЕ ОБУЧЕНИЕ В ИДЕНТИФИКАЦИИ СПИКЕРА: СОВРЕМЕННЫЕ МЕТОДЫ И ПЕРСПЕКТИВЫ РАЗВИТИЯ.....	98

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

Д. Лукьянов, А. Колесников ИСПОЛЬЗОВАНИЕ ЭНТРОПИЙНОГО ПОДХОДА В АНАЛИЗЕ СИСТЕМ ЗНАНИЙ В СФЕРЕ УПРАВЛЕНИЯ ПРОЕКТАМИ НА ПРИМЕРЕ ISV 4.0 IPMA	110
П.С. Пустовойтов, Н.А. Сейлова, А.С. Гнатюк МЕТОДОВ ПОДДЕЛКИ ГОЛОСА: РИСКИ, СЛУЧАИ И СТРАТЕГИИ ЗАЩИТЫ.....	122

CONTENTS

DIGITAL TECHNOLOGIES IN THE DEVELOPMENT OF SOCIO-ECONOMIC SYSTEMS

A. Agdavletova, V. Madin, O. Salykova ADAPTIVE PROCESS MANAGEMENT USING DEEP LEARNING ON A PROGRAMMABLE LOGIC CONTROLLER (PLC).....	8
F. Bhat, N.A. Seilova, V.V. Pokusov SOFTWARE IMPLEMENTATION OF TESTING METHODOLOGY AND LOAD TESTING OF COMPUTERS.....	29
Zh.M. Doskhozina THE PRINCIPLES OF INTERCULTURAL COMMUNICATION IN THE MODERN WORLD.....	48
U.R. Yeraliev THE MILITARY SPHERE AS A COMPONENT OF INTERNATIONAL RELATIONS.....	56
A. Tuleubekov, A. Doskozhanova ON CONTEMPORARY TECHNOLOGIES: HEIDEGGER'S APPROACH.....	63

INFORMATION TECHNOLOGY

A. Maratuly, Y.A. Abibullayev PERFORMANCE STUDY AND COMPARATIVE ANALYSIS OF YOLO-NAS AND PREVIOUS VERSIONS OF YOLO.....	71
Y.Y. Muratkhanov, Y.A. Zhanbabayev IMPORTANCE OF IT-TECHNOLOGIES IN CREATING OF INTERNATIONAL RELATIONSHIPS.....	84
K.A. Shertayev, L.K. Naizabayeva DEEP LEARNING IN SPEAKER IDENTIFICATION: MODERN METHODS AND DEVELOPMENT PROSPECTS.....	98

INFORMATION SECURITY AND COMMUNICATION TECHNOLOGIES

D. Lukianov, O. Kolesnikov USING THE ENTROPY APPROACH IN THE ANALYSIS OF KNOWLEDGE SYSTEMS IN THE FIELD OF PROJECT MANAGEMENT BY THE EXAMPLE OF ICB 4.0 IPMA	110
P.S. Pustovoitov, N.A. Seilova, A.S. Gnatiuk ANALYSIS OF VOICE IMPERSONATION FRAUD: RISKS, CASES AND DEFENSE STRATEGIES.....	122

INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

ISSN 2708–2032 (print)

ISSN 2708–2040 (online)

Vol. 5. Is. 1. Number 17 (2024). Pp. 29–47

Journal homepage: <https://journal.iitu.edu.kz>

<https://doi.org/10.54309/IJICT.2024.17.1.002>

UDK 004

SOFTWARE IMPLEMENTATION OF TESTING METHODOLOGY AND LOAD TESTING OF COMPUTERS

F. Bhat¹, N.A. Seilova^{1}, V.V. Pokusov²*

¹International Information Technology University, Almaty, Kazakhstan;

²Association of Legal Entities «Kazakhstan Information Security Association»,
Astana, Kazakhstan.

E-mail: nseilova@iitu.edu.kz

Bkhat Fardin — master student, Faculty of Computer Technologies and Cyber Security, International Information Technology University, Almaty, Kazakhstan

E-mail: 36171@iitu.edu.kz. ORCID: 0009-0004-4692-4644;

Seilova Nurgul — candidate of technical sciences, associate professor of the Department of Computer Engineering, International Information Technology University, Almaty, Kazakhstan

E-mail: nseilova@iitu.edu.kz. ORCID: 0000-0003-3827-179X;

Pokusov Victor Vladimirovich — chairman of Association of Legal Entities «Kazakhstan Information Security Association», Astana, Kazakhstan

ORCID: 0000-0002-5251-3452.

© F. Bhat, N.A. Seilova, V.V. Pokusov, 2024

Abstract. The authors of the article propose a methodology for testing and load testing of computers with an emphasis on creating a graphical user interface (GUI) and implementing a monitoring system. Two key components of the methodology include: a computer component detection module and a computer performance monitoring system. The component detection module provides an automated mechanism for collecting information about computer components from various sources, allowing for more accurate and relevant testing in different configurations. The implementation of a graphical user interface for this module simplifies user interaction with the system, providing convenient means of interaction and visualization of results. The monitoring system provides continuous monitoring of the computer status in real time. This includes collecting, analyzing and visualizing performance data to enable rapid response to changes and provide greater insight into system performance during testing. Particular attention is paid to data processing, ensuring stability and visualization of results. This serves as a key resource for developers and computer testing professionals, providing a detailed overview of the practical implementation of the tools needed to effectively test and monitor modern computer systems.



Keywords: performance, load testing, monitoring system, graphical user interface, computer hardware

For citation: F. Bhat, N.A. Seilova, V.V. Pokusov. SOFTWARE IMPLEMENTATION OF TESTING METHODOLOGY AND LOAD TESTING OF COMPUTERS// INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES. 2024. Vol. 5. No. 17. Pp. 29–47 (In Russ.). <https://doi.org/10.54309/IJICT.2024.17.1.002>.

КОМПЬЮТЕРЛЕРДІ ЖҮКТЕМЕЛІК ТЕСТІЛЕУ БОЙЫНША СЫНАҚТАР ӘДІСТЕМЕСІН ЖҮРГІЗУ БАҒДАРЛАМАСЫН ЖҮЗЕГЕ АСЫРУ

Ф. Бхат¹, Н.А. Сейлова^{1}, В.В. Покусов²*

¹Халықаралық ақпараттық технологиялар университеті, Алматы, Қазақстан;

²Қазақстанның ақпараттық қауіпсіздік қауымдастығы, Астана, Қазақстан.

E-mail: nseilova@iitu.edu.kz

Бхат Фардин — Халықаралық ақпараттық технологиялар университетінің компьютерлік технологиялар және киберқауіпсіздік факультетінің магистрі, компьютерлік технологиялар және бағдарламалық қамтамасыз ету саласында маманданған

E-mail: 36171@iitu.edu.kz. ORCID: 0009-0004-4692-4644;

Сейлова Нургуль Абадуллаевна — техника ғылымдарының кандидаты, компьютерлік инженерия кафедрасының доценті, Халықаралық ақпараттық технологиялар университеті

E-mail: nseilova@iitu.edu.kz. ORCID: 0000-0003-3827-179X;

Покусов Виктор Владимирович — «Қазақстанның ақпараттық қауіпсіздік қауымдастығы» заңды тұлғалар бірлестігінің төрағасы

ORCID: 0000-0002-5251-3452.

© Ф. Бхат, Н.А. Сейлова, В.В. Покусов, 2024

Аннотация. Ақпараттық технологиялар мен компьютерлік жүйелер саласындағы заманауи үрдістер негізінде графикалық пайдаланушы интерфейсі (GUI) құруға және мониторинг жүйесін іске асыруға баса назар аудара отырып, компьютерлерді сынау және жүктемелік тестілеу әдістемесі ұсынылады. Осы зерттеуді "іске асыру" барысында Әдістеменің екі негізгі компонентін әзірлеу және енгізу егжей-тегжейлі сипатталған: компьютердің компоненттерін анық тау модулі және оның жұмысын бақылау жүйесі. Компоненттерді анықтау модулі әртүрлі көздерден компьютердің құрамдас бөліктері туралы ақпаратты жинаудың автоматтандырылған механизмін қамтамасыз етеді, бұл әртүрлі конфигурацияларда тестілеуді дәлірек және өзекті етуге мүмкіндік береді. Берілген модуль үшін графикалық интерфейсін енгізу пайдаланушының жүйемен өзара әрекеттесуін жеңілдетеді, өзара әрекеттесу мен нәтижелерді визуализациялаудың ыңғайлы құралдарын ұсынады. Мониторинг жүйесі, өз кезегінде, нақты уақыт режимінде компьютердің күйін үнемі бақылауда ұстайды. Бұған өнімділік деректерін жинау, талдау және визуализация кіреді, бұл өзгерістерге жедел жауап



беруге мүмкіндік береді және тестілеу процесінде жүйенің жұмысын тереңірек түсінуге мүмкіндік береді. "Іске асыру" бөлімі технология мен архитектураны таңдаудан бастап негізгі тестілеу жүйесімен интеграция туралы егжейіне дейін осы компоненттерді құрудың техникалық шолуын ұсынады. Деректерді өңдеуге, тұрақтылықты қамтамасыз етуге және нәтижелерді визуализациялауға ерекше назар аударылады. Бұл тарау зерттеушілер, әзірлеушілер және компьютерлік тестілеу мамандары үшін негізгі ресурс ретінде қызмет етеді, қазіргі заманғы компьютерлік жүйелерді тиімді тестілеу және бақылау үшін қажетті құралдарды практикалық енгізуге толық шолу жасайды.

Түйін сөздер: өнімділік, жүктемені тестілеу, бақылау жүйесі, графикалық пайдаланушы интерфейсі, компьютерлік жабдық

Дәйексөздер үшін: Ф. Бхат, Н.А. Сейлова, В.В. Покусов. КОМПЬЮТЕРЛЕРДІ ЖҮКТЕМЕЛІК ТЕСТІЛЕУ БОЙЫНША СЫНАҚТАР ӨДІСТЕМЕСІН ЖҮРГІЗУ БАҒДАРЛАМАСЫН ЖҮЗЕГЕ АСЫРУ// ХАЛЫҚАРАЛЫҚ АҚПАРАТТЫҚ ЖӘ-НЕ КОММУНИКАЛЫҚ ТЕХНОЛОГИЯЛАР ЖУРНАЛЫ. 2024. Т. 5. №. 17. 29–47 бет. (орыс тіл.). <https://doi.org/10.54309/IJICT.2024.17.1.002>.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДИКИ ПРОВЕДЕНИЯ ИСПЫТАНИЙ И НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ КОМПЬЮТЕРОВ

Ф. Бхат¹, Н.А. Сейлова^{1}, В.В. Покусов²*

¹Международный университет информационных технологий, Алматы, Казахстан;

²ОЮЛ «Казахстанская Ассоциация Информационной Безопасности», Астана, Казахстан.

E-mail: nseilova@iitu.edu.kz

Бхат Фардин — магистрант, факультет компьютерных технологий и кибербезопасности, Международный университет информационных технологий, Алматы Казахстан
E-mail: 36171@iitu.edu.kz. ORCID: 0009-0004-4692-4644;

Сейлова Нургуль Абадуллаевна — к.т.н., ассоциированный профессор кафедры компьютерная инженерия, Международный университет информационных технологий, Алматы Казахстан
E-mail: nseilova@iitu.edu.kz. ORCID: 0000-0003-3827-179X;

Покусов Виктор Владимирович — председатель ОЮЛ «Казахстанская Ассоциация Информационной Безопасности», Астана, Казахстан
ORCID: 0000-0002-5251-3452.

© Ф. Бхат, Н.А. Сейлова, В.В. Покусов, 2024

Аннотация. На основе современных тенденций в области информационных технологий и компьютерных систем, предлагается методика проведения испытаний и нагрузочного тестирования компьютеров с акцентом на создание графического пользовательского интерфейса (GUI) и реализацию системы мониторинга. Авторы подробно описывают разработку и внедрение двух ключевых компонентов методики: модуля определения комплектующих компьютера и системы мониторинга его производительности. Модуль определения комплектующих обеспечивает



автоматизированный механизм сбора информации о составляющих компьютера из различных источников, что позволяет более точно проводить тестирование в различных конфигурациях. Реализация графического пользовательского интерфейса для данного модуля упрощает взаимодействие пользователя с системой, предоставляя удобные средства взаимодействия и визуализацию результатов. Система мониторинга, в свою очередь, обеспечивает постоянное наблюдение за состоянием компьютера в режиме реального времени. Это включает в себя сбор, анализ и визуализацию данных о производительности, что позволяет оперативно реагировать на изменения и обеспечивает более глубокое понимание работы системы в процессе проведения тестов. В статье дается технический обзор создания этих компонентов, начиная от выбора технологий и архитектуры, заканчивая деталями по интеграции с основной системой тестирования. Особое внимание уделяется обработке данных, обеспечению стабильности и визуализации результатов, что служит ключевым ресурсом для исследователей, разработчиков и специалистов в области тестирования компьютеров, предоставляя подробный обзор практической реализации инструментария, необходимого для эффективного тестирования и мониторинга современных компьютерных систем.

Ключевые слова: производительность, нагрузочное тестирование, система мониторинга, графический пользовательский интерфейс, компьютерное оборудование

Для цитирования: Ф. Бхат, Н.А. Сейлова, В.В. Покусов. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДИКИ ПРОВЕДЕНИЯ ИСПЫТАНИЙ И НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ КОМПЬЮТЕРОВ//МЕЖДУНАРОДНЫЙ ЖУРНАЛ ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ. 2024. Т. 5. No. 17. Стр. 29–47. (На русс.). <https://doi.org/10.54309/IJICT.2024.17.1.002>.

Введение

Современная зависимость от информационных технологий требует обеспечения стабильности, надежности и высокой производительности компьютерных систем. В контексте этой задачи, проведение испытаний и нагрузочного тестирования становится неотъемлемым этапом, направленным на выявление и оптимизацию производственных характеристик аппаратных компонентов. Однако, эффективная реализация данных методик требует комплексного и системного подхода, включая использование специализированных программных инструментов.

В данной работе фокус находится на создании программного обеспечения, предназначенного для упрощения и автоматизации процесса испытаний и нагрузочного тестирования компьютеров (Afef et al., 2015). Принимая во внимание современные требования, связанные с многозадачностью и сложностью аппаратных конфигураций (Randal et al., 2015), было решено использовать основные средства, встроенные в алгоритмический код языка программирования, а также функциональность библиотек операционной системы Windows 10. Это стратегическое решение направлено на обеспечение оптимальной совместимости с современным программным окружением и максимизацию эффективности



проводимых тестирований. Таким образом, представленное программное обеспечение представляет собой инновационный подход к испытаниям компьютерных систем, обеспечивая точность, надежность и высокую производительность в современных условиях информационных технологий.

Процесс имплементации системы мониторинга (Lazareva et al., 2020), включает в себя сбор данных с определенных компонентов с прошлого этапа, с последующим анализом характеристик устройства в определенный период времени или на протяжении довольно длительного процесса работы. Эта система мониторинга охватывает различные аспекты работы устройства, такие как производительность процессора, использование оперативной памяти, нагрузка на жесткий диск, температурные параметры и другие характеристики. Такой комплексный анализ позволяет оперативно выявлять потенциальные проблемы, отслеживать изменения в работе системы, а также принимать предупредительные меры для предотвращения сбоев или ухудшения производительности (Turovec et al., 2022).

Для разработки программного обеспечения информационная система нагрузочного тестирования была разделена на несколько компонентов. В результате декомпозиции были выставлены следующие системные задачи: Определение комплектующих компьютера, Определение системы мониторинга, Определение системы нагрузочного тестирования, Определение графического пользовательского интерфейса.

Материалы и методы

Данный параграф включает в себя несколько этапов в разработке собственной информационной системы для нагрузочного тестирования. Ниже представлен детальный обзор материалов, используемых в ходе реализации, и методов, применяемых для достижения поставленных целей.

Язык программирования

Исходя из второй главы исследовательской работы, посвященной проектированию, было принято стратегическое решение в пользу языка программирования Pure Basic. Этот выбор обусловлен рядом ключевых преимуществ, сделавших данный язык идеальным инструментом для реализации нашей методики испытаний и нагрузочного тестирования компьютеров. Прежде всего, стоит выделить высокую производительность этого языка, что крайне важно для разработки программ, требующих максимальной эффективности выполнения кода. Вторым существенным преимуществом является портативность, позволяющая без проблем переносить наше программное обеспечение между различными аппаратными конфигурациями. Кроме того, автономность от дополнительных библиотек делает Pure Basic независимым и облегчает управление и развертывание тестовых стендов и приложений.

Система нагрузочного тестирования

В ходе разработки информационной системы используются основные средства, встроенные в алгоритмический код языка программирования, а также функциональность библиотек операционной системы Windows 10. Этот подход позволил эффективно использовать возможности языка программирования,



обеспечивая гибкость и оптимизацию работы системы. Встроенные функции языка программирования предоставили широкий спектр инструментов для обработки данных, взаимодействия с пользовательским интерфейсом и другими ключевыми аспектами разработки. В дополнение к этому, использование библиотек операционной системы Windows 10 обеспечивала стабильность и совместимость с окружающей средой пользователя, что являлась важным фактором для эффективной работы информационной системы в данном окружении. Выбор данного метода хорошо оптимизировало процесс разработки программного обеспечения, поскольку использование встроенных средств позволило избежать необходимости создания сложных алгоритмов с нуля, и сосредоточить усилия на решении основной задачи тестовой системы.

Графический пользовательский интерфейс

Система графического пользовательского интерфейса представляет собой неотъемлемую часть, которая должна обладать высокой степенью интуитивности для пользователей. Она включает в себя разрабатываемое программное обеспечение, предоставляющее удобный и понятный доступ к информации о комплектующих пользовательского персонального компьютера. Важным аспектом этой системы является наличие заранее собранной и структурированной информации о компонентах компьютера, обеспечивая пользователя всей необходимой и актуальной информацией о состоянии системы. Кроме того, система графического пользовательского интерфейса взаимодействует с имплементированной системой мониторинга, предоставляя пользователю удобный доступ к данным о производительности, нагрузке и других важных параметрах.

Сбор информации о комплектующих компьютера

Данный процесс включает в себя подробный анализ аппаратной составляющей, такой как материнская плата, процессор, жесткий диск, оперативно запоминающее устройство (ОЗУ) и другие устройства. Важно учесть технические параметры, такие как скорость процессора, объем оперативной памяти, емкость дискового пространства и интерфейсы подключения. В процессе разработки компонента сбора информации присутствует несколько вариантов выбора методов захвата аппаратной информации. Этот этап представляет важное решение, поскольку от выбора методологии зависит эффективность и точность собираемых данных. Рассматриваются различные техники, включая использование стандартных API операционной системы, взаимодействие с драйверами устройств, а также анализ данных, предоставляемых BIOS компьютера. Каждый из этих подходов имеет свои преимущества и ограничения, и выбор осуществляется с учетом конкретных требований проекта. В данной реализации используется оптимальный метод с учетом баланса между точностью данных, производительностью и совместимостью с различными конфигурациями оборудования. Этот комплексный процесс обеспечивает систему нагрузочного тестирования надежными и актуальными данными об аппаратной структуре компьютера, что в свою очередь содействует эффективному проведению тестов и более точной оценке производительности системы (Utkin et al., 2009) в условиях повышенной нагрузки.



Сбор информации через реестр Microsoft Windows

Реестр представляет собой иерархически построенную базу данных, в которой хранятся параметры и настройки, касающиеся аппаратного обеспечения, программного обеспечения, профилей пользователей и предустановленных компонентов. В нем фиксируются большинство изменений, совершаемые в системе, такие как модификации в настройках Панели управления, ассоциации файлов, системные политики, а также перечень установленного программного обеспечения. Поддерживая централизованный доступ к этим данным, реестр обеспечивает системе операционной системы средство эффективного управления параметрами, что важно для обеспечения стабильности и производительности компьютерной системы.

На рисунке 1, изображена программная реализация пользовательской функции сбора информации из Реестра Windows. Была собрана информация, касающаяся процессора, а именно параметров имени и вендора производителя процессора.

```

; =====
; Возможен баг, если заменить процессор, то информация о процессоре не обновится (2007 год)
|Procedure.s RegReadString(section, path$, key$)
|
| #KEY_READ = 131097
| #ERROR_SUCCESS = 0
| #HKEY_LOCAL_MACHINE = 2147483650
|
| value.s = Space(9999) : datasize.l = 255
| If RegOpenKeyEx(section, path$, 0, #KEY_READ, @tmp) = #ERROR_SUCCESS
|   If RegQueryValueEx(tmp, key$, 0, 0, @value, @datasize) <> #ERROR_SUCCESS : value="-1" : EndIf
|   RegCloseKey(tmp)
| EndIf
| ProcedureReturn value
| EndProcedure

ProcessorLoc.s = "HARDWARE\DESCRIPTION\System\CentralProcessor\0"

Debug "Processor name: " + Trim(RegReadString(#HKEY_LOCAL_MACHINE, ProcessorLoc, "ProcessorNameString")); Processor name
Debug "Processor vendor: " + Trim(RegReadString(#HKEY_LOCAL_MACHINE, ProcessorLoc, "VendorIdentifier")); Vendor name
; =====

```

Рисунок 1 – «Функция получения данных из Реестра Windows»

На рисунке 2, представлены данные отражающий запрошенные характеристики центрального процессора, такие как его наименование и производитель.

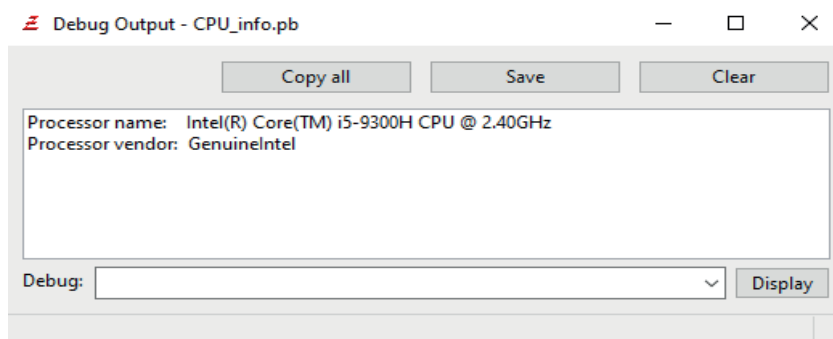


Рисунок 2 – «Вывод информации из реестра Windows»

Часть собираемой информации из реестра Windows представляет собой данные, которые аккумулируются после первой установки операционной системы Microsoft Windows. Реестр служит центральным хранилищем, где хранятся ключевые данные, необходимые для корректного функционирования Windows. Извлечение данных из хранилища предоставляет исчерпывающий обзор системы, включая ее настройки, программное обеспечение и аппаратную конфигурацию. Однако информация, которая хранится в реестре, может быть, не актуальна, так как во время разработки с помощью данного метода была замечена неприемлемое несоответствие. Так как реестр Windows устанавливается вместе с операционной системой Windows, данные, которые собираются и записываются в реестр происходят на этапе установки, что приводит к тому, что система на некоторых моделях материнских плат запоминает процессор, который был установлен во время установки операционной системы. И в случае, когда процессор будет заменен, есть шанс того, что информация о новом центральном процессоре внутри реестра не обновится, что приведет к проблемам во время сбора информации.

Метод сбора информации из реестра Microsoft Windows, хоть и предоставляет ценные данные, становится менее актуальным в контексте переустановки или модификации аппаратного оборудования. Это обусловлено тем, что при таких изменениях информация в реестре не всегда обновляется автоматически. Тем не менее, этот метод продолжает быть полезным для сбора информации о программном обеспечении и измененных данных на уровне пользовательского интерфейса.

Сбор информации через WIN32 API

В процессе разработки были задействованы два различных метода получения данных с оборудования, основанных на WIN API (Magapules, 2019). Первый подход включал использование функций, которые обращаются к программно-аппаратным средствам, позволяя осуществлять более непосредственный доступ к характеристикам оборудования, хоть порой и не очень подробный. Второй вариант включал в себя использование WMI (Windows Management Instrumentation), предоставляя более абстрактный уровень доступа к системным ресурсам. WMI предоставляет интерфейс для взаимодействия с данными, представленными в формате CIM (Common Information Model), внутри операционной системы Windows. WMI включает в себя реализацию стандарта CIM для Windows систем, как программный интерфейс для управления компонентами операционной системы. Оба метода имеют свои преимущества и применимы в зависимости от конкретных требований и задач в различных частях информационной системы.

В WIN32 информация об аппаратном оборудовании находится в специально выделенном компоненте называемым WMI, это одна из фундаментальных технологий, предназначенных для централизованного контроля и мониторинга деятельности различных компонентов компьютерной инфраструктуры, работающих под управлением операционной системы Windows. WMI включает в себя реализацию стандарта CIM для Windows систем, как программный интерфейс для управления компонентами операционной системы.



На уровне программирования эта концепция была воплощена через создание пользовательской функции, которая позволяет отправлять запросы с указанными параметрами. Реализация такого функционала способствует более гибкому и контролируемому взаимодействию с системными ресурсами через программный код. На рисунке 3 приведен скриншот программного кода, реализованной функции.

```

;=====
Procedure s CIM_Sys_Name(CIM.s, CIM_param.s)
Define .COMateObject objWMIService, SysName
colSysNames.COMateEnumObject
strComputer.s = "."

objWMIService = COMate_GetObject("winmgmts:\\" + strComputer + "\root\cimv2", "")
If objWMIService
colSysNames = objWMIService\CreateEnumeration("ExecQuery('Select * from " + CIM + "')")
If colSysNames
SysName= colSysNames\GetNextObject()
While SysName
System_Name.s = SysName\GetStringProperty(CIM_param)
SysName\Release()
SysName = colSysNames\GetNextObject()
Wend
colSysNames\Release()
EndIf
objWMIService\Release()
EndIf
ProcedureReturn System_Name
EndProcedure
;=====

```

Рисунок 3 – «Функция отправки данных в WMI»

Приведенная выше функция принимает аргументы от различных пользовательских функций (к примеру функции получения данных процессора «getCPU()»), и отправляет проработанные данные из WMI обратно в виде ответа возвращаемого значения процедуры.

Функция получения информации о процессоре

На рисунке 4 изображена функция получения и обработки данных о процессоре посредством отправки запроса в WMI.

```

;=====
Procedure getCPU()

#CIM_Processor = "CIM_Processor"
#CPU_Name = "Name"
#CPU_Manufacturer = "Manufacturer"
#CPU_Socket = "SocketDesignation"
#CPU_Load = "LoadPercentage"
#CPU_Cores = "NumberOfCores"
#CPU_ThreadCount = "ThreadCount"
#CPU_CurrentClockSpeed = "CurrentClockSpeed"

Global Name.s = CIM_Sys_Name(#CIM_Processor, #CPU_Name)
Global Manufacturer.s = CIM_Sys_Name(#CIM_Processor, #CPU_Manufacturer)
Global Socket.s = CIM_Sys_Name(#CIM_Processor, #CPU_Socket)
Global Cores.s = CIM_Sys_Name(#CIM_Processor, #CPU_Cores)
Global Threads.s = CIM_Sys_Name(#CIM_Processor, #CPU_ThreadCount)
Global CurrentClockSpeed.s = CIM_Sys_Name(#CIM_Processor, #CPU_CurrentClockSpeed)

EndProcedure
;=====

```

Рисунок 4 – «Функция регистрации запросов и обработки ответов для процессора»

На рисунке 5 представлен вывод, отражающий запрошенные характеристики центрального процессора из WMI, такие как его имя, производитель, сокет материнской платы для процессора, количество ядер, количество потоков и текущую скорость процессора.

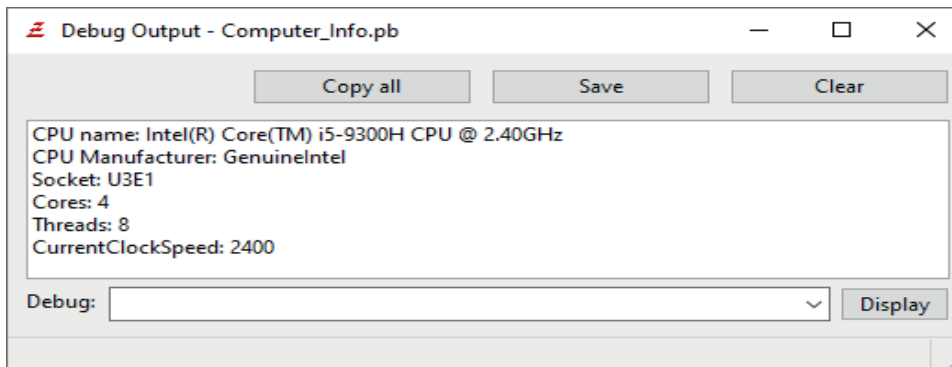


Рисунок 5 – «Вывод информации о процессоре»

Алгоритм получения данных из других комплектующих компьютера осуществляется по такому же принципу, за исключением того, что для каждого компонента реализована отдельная пользовательская функция, отправляющая в WMI запрос с определенными атрибутами, касающимися запрашиваемого оборудования. Например, для получения информации о процессоре используется функция, которая формирует запрос к WMI с учетом основных характеристик процессора, таких как тактовая частота, количество ядер и потоков. Аналогичные функции реализованы для сбора данных об оперативной памяти, графической карте, жестких дисках и других ключевых компонентах.

Это модульное устройство алгоритма обеспечивает гибкость и расширяемость системы, позволяя легко добавлять и обновлять функциональность для новых типов оборудования. Такой подход к сбору данных об аппаратных компонентах компьютера обеспечивает не только высокую точность и надежность данных, но и легкость адаптации к изменениям в технологическом прогрессе и появлению новых видов компьютерных устройств. Тем самым, разработанное программное обеспечение представляет собой эффективный и современный инструмент для проведения нагрузочных тестов и оценки производительности компьютерных систем.

Функция получения информации о материнской плате и BIOS

Для получения данных о материнской плате компьютера была создана отдельная функция изображенная на рисунке 6. Процедура «getMainboard()» предназначена для регистрации запросов о материнской плате для последующего отправления в функцию «CIM_Sys_Name()». В данной функции создаются глобальные переменные хранящие полученные результаты из функции «CIM_Sys_Name()».



Также в данном отрезке кода была получена информация о микросхеме BIOS, которая расположена на материнской плате каждого компьютера.

Базовая система ввода-вывода (BIOS) представляет собой фундаментальный компонент компьютерной системы, ответственный за основные функции во время её загрузки. Этот встроенный программный модуль хранится в постоянной памяти, обычно на микросхеме флэш-памяти на материнской плате. Внутри BIOS содержится набор низкоуровневых инструкций и параметров, которые необходимы для инициализации и проверки аппаратных компонентов при включении компьютера. Среди ключевых данных, которые хранит BIOS, входят информация о конфигурации железа, такая как типы и характеристики процессоров, памяти и периферийных устройств. Кроме того, BIOS включает в себя таблицу разделов жёсткого диска и базовые параметры системного времени и даты. Эти данные играют важную роль в стабильной и эффективной работе материнской платы и обеспечивают необходимую связь между аппаратным обеспечением и программным обеспечением компьютера.

```

;=====
Procedure getMainboard()
;==== Motherboard =====
#CIM_Mainboard = "Win32_BaseBoard"
#Mainboard_Manufacturer = "Manufacturer"
#Mainboard_Model = "Product"
Global MB_Manufacturer.s = CIM_Sys_Name(#CIM_Mainboard, #Mainboard_Manufacturer)
Global MB_Model.s = CIM_Sys_Name(#CIM_Mainboard, #Mainboard_Model)

;==== BIOS =====
#CIM_BIOS = "Win32_BIOS"
#BIOS_Version = "SMBIOSBIOSVersion"
#BIOS_Brand = "Manufacturer"
#BIOS_Date = "ReleaseDate"
;#BIOS_SN = "SerialNumber"

Global BIOSBrand.s = CIM_Sys_Name(#CIM_BIOS, #BIOS_Brand)
Global BIOSVersion.s = CIM_Sys_Name(#CIM_BIOS, #BIOS_Version)
BIOSDate_RAW.s = CIM_Sys_Name(#CIM_BIOS, #BIOS_Date)
Global BIOSDate.s = RemoveString(BIOSDate_RAW, "0", #PB_String_CaseSensitive, 6)
BIOSDate = RemoveString(BIOSDate, ".")
BIOSDate = RemoveString(BIOSDate, "+")

EndProcedure
;=====

```

Рисунок 6 – «Функция регистрации запросов и обработки ответов для материнской платы»

На рисунке 7, представлены данные, полученные в результате компиляции программы. В частности, на изображении представлен вывод, отражающий запрошенные характеристики материнской платы и BIOS. Из материнской платы извлекли информацию касающуюся модели и производителя материнской платы. Из флэш памяти BIOS были получены данные о версии, брэнда и дате установки BIOSа.

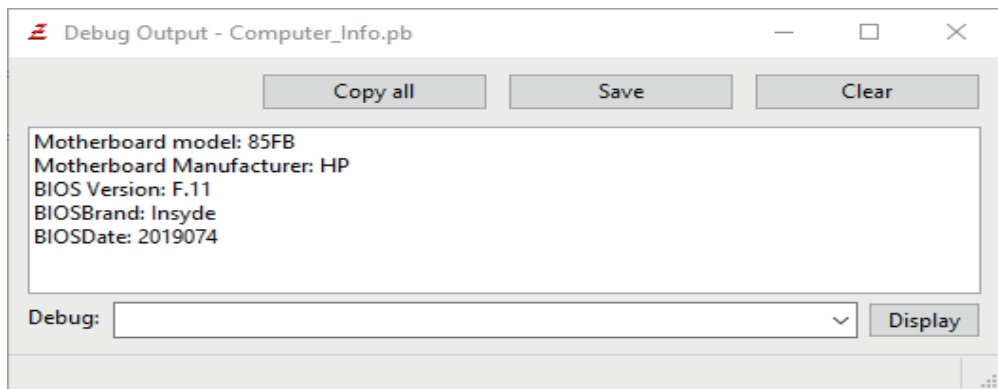


Рисунок 7 – «Вывод информации о материнской плате и BIOS»

Функция получения информации об Оперативно Запоминающем Устройстве

Для сбора информации о Оперативно Запоминающем Устройстве компьютера была разработана отдельная функция, иллюстрированная на рисунке 8. В рамках данной функции создаются глобальные переменные, предназначенные для хранения результатов, полученных из функции «CIM_Sys_Name()». В процессе выполнения функции осуществляется запись собранной информации для последующего использования в других функциях.

В данной функции так же был реализован новый метод сбора информации, данные об использовании оперативной памяти. Для получения загруженности ОЗУ, была создана структура данных «memex», которая содержит сведения о текущем состоянии физической и виртуальной памяти, предоставляемыми системными службами операционной системы. Внутри структуры мы обратились к части «dwMemoryLoad», которая в свою очередь содержит информацию о проценте используемой физической памяти.

```

;=====
Procedure getRAM()
;==== RAM % ====
memex.MEMORYSTATUSEX
memex\dwLength = SizeOf(MEMORYSTATUSEX)
GlobalMemoryStatusEx(@memex)
Global RAM_usage.s = Str(memex\dwMemoryLoad) + "%"

;==== RAM info ====
#CIM_RAM = "CIM_PhysicalMemory"
#RAM_Manufacturer = "Manufacturer"
#RAM_Speed = "Speed"
#RAM_Capacity = "Capacity"
Global RAMManufacturer.s = CIM_Sys_Name(#CIM_RAM, #RAM_Manufacturer)
Global RAMSpeed.s = CIM_Sys_Name(#CIM_RAM, #RAM_Speed)
Global RAMCapacity.s = CIM_Sys_Name(#CIM_RAM, #RAM_Capacity)
Global RAMMemoryALL.s = Str(MemoryStatus(#PB_System_TotalPhysical) / 1023 / 1023)

EndProcedure
;=====

```

Рисунок 8 – «Функция регистрации запросов и обработки ответов для ОЗУ»



На рисунке 9 представлены сведения, полученные в результате компиляции программы. В частности, на данном графическом представлении отображается результат выполнения, предоставляющий информацию о характеристиках Оперативно Запоминающего Устройства (ОЗУ). Были извлечены данные, относящиеся к модели, частоте и объему оперативной памяти.

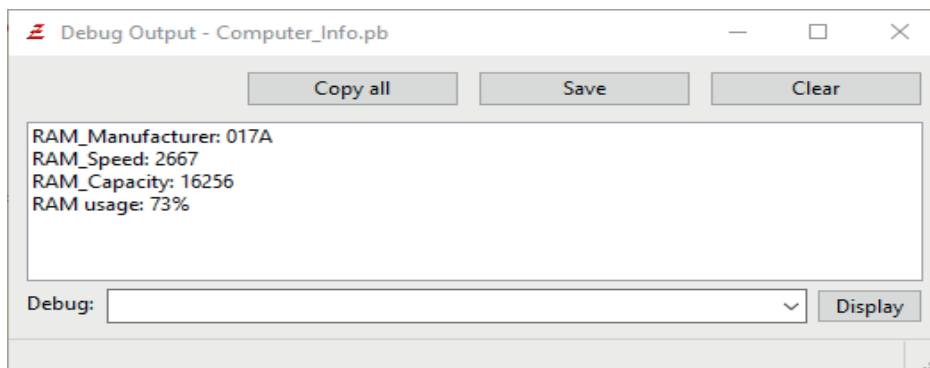


Рисунок 9 – «Вывод информации об Оперативно Запоминающем Устройстве»

Функция получения информации о жестких дисках

Для сбора информации о зарегистрированных жестких дисках было принято решение воспользоваться встроенными функциями операционной системы. При разработке были задействованы различные методы, включая, например, функцию «GetDriveType()». Эта функция позволяет получить полезные сведения о типе жесткого диска, включая информацию о том, является ли диск съемным, фиксированным, компакт-диском, диском ОЗУ или сетевым диском.

Кроме того, для получения подробной информации о доступном пространстве на дисках и общем объеме памяти была задействована системная функция «GetDiskFreeSpaceEx()». Эта функция предоставляет данные о свободном и занятом пространстве на дисках, что включает в себя как общий объем памяти, так и количество доступного места.

Таким образом, на рисунке 10 изображена программная реализация в виде пользовательской функции, которая включает в себя разносторонний подход к сбору информации о жестких дисках, обеспечивая более полное представление о характеристиках системы хранения данных.

Процедура «SSDHDDInfo()» внутри поделена абстрактно поделена на три крупных блока кода. В первой части происходит инициализация ресурсов и данных, с которыми будет производиться действие, объявление типов жестких дисков, переменных, которые хранят информацию с последующей передачи их в системные функции. Вторая часть кода включает в себя циклический алгоритм, определяющий количество носителей информации, тип жесткого диска и атрибут наименования диска в системе. В третьем блоке кода был имплементирован алгоритм получения информации о размере, и о оставшимся месте на диске.

```

;
Procedure SSDHDDInfo() ;Получение информации о дисках

Enumeration
#DRIVE_UNKNOWN
#DRIVE_NO_ROOT_DIR
#DRIVE_REMOVABLE
#DRIVE_FIXED
#DRIVE_REMOTE
#DRIVE_CDROM
#DRIVE_RAMDISK
EndEnumeration

Define q BytesFreeToCaller, TotalBytes, TotalFreeBytes
i.b = 0
Dim drives.s(@i) ; размер массива динамически изменяется

;===== ПОИСК ДИСКА =====
For vol=65 To 90:
vol$ = Chr(vol)+":\\"
If GetDriveType_(vol$) = #DRIVE_FIXED
Debug vol$+" is a hard drive"
drives(i) = vol$
i=i+1
ElseIf GetDriveType_(vol$) = #DRIVE_REMOVABLE
Debug vol$+" is a removable drive"
drives(i) = vol$
i=i+1
EndIf
Next

;===== СБОР ПАМЯТИ С ДИСКОВ =====
For volume = 0 To (i-1):
If GetDiskFreeSpaceEx_(@drives(volume), @BytesFreeToCaller, @TotalBytes, @TotalFreeBytes) = 0
MessageRequester("", "Drive not ready!", 0)
End
EndIf

Debug "Disk: " + drives(volume)
Debug "Size: " + FormatNumber(TotalBytes/ (1024*1024), 0) + " GB"
Debug "Free: " + FormatNumber(TotalFreeBytes/ (1024*1024), 0) + " GB"
Next

EndProcedure
;=====

```

Рисунок 10 – «Вывод информации об Оперативно Запоминающем Устройстве»

В представленном на рисунке 11 графическом отчёте представлены результаты обработки данных после вызова функции «SSDHDDInfo()». В выводе данного отчета производится перечисление зарегистрированных накопителей информации с указанием их типа, различных атрибутов, таких как имя диска, а также информации об общем объеме памяти и доступном пространстве на каждом жестком диске.

Так же в выводе можно заметить, что информационные накопители «C:\» и «D:\» являются системными, и то, что они относятся к типу жестких дисков, тогда как диск «E:\» это флэш накопитель, вставленный в порт компьютера, и помечается с типом - съемный диск.

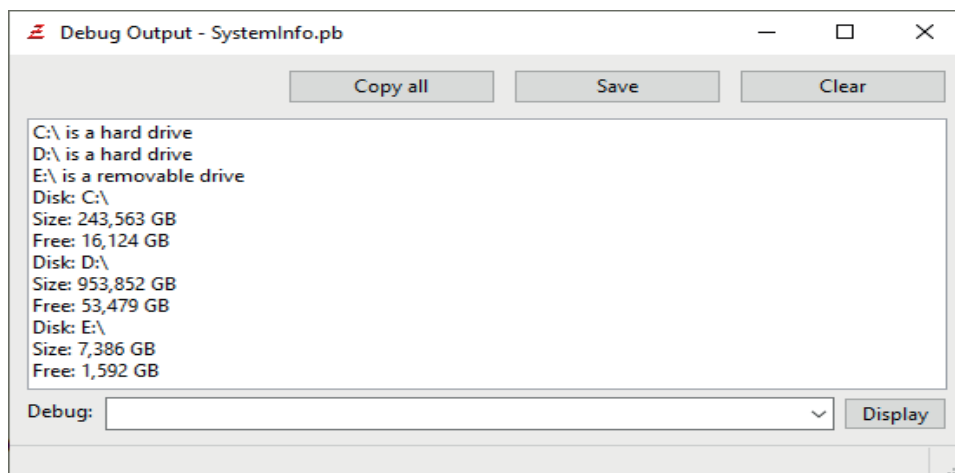


Рисунок 11 – «Вывод информации об жестких дисках»

Результаты

Прототип пользовательского приложения

В завершении этапа сбора информации начинается фаза агрегации данных о комплектующих компьютера. В процессе разработки программного обеспечения предпринимались усилия по оптимальному использованию функционала, предоставляемого операционной системой. Осуществленный выбор определял точечное внимание реализации графического пользовательского интерфейса, с чем было решено воспользоваться функционалом библиотеки WIN API (Korablin Yu et al., 2013). Этот выбор обосновывался стремлением к созданию интерфейса, который выделяется интуитивной понятностью и удобством использования, что совместно обеспечивает эффективное взаимодействие пользователя с приложением. Используя встроенные средства и библиотеку WIN API, достигнута успешная интеграция собранных данных, а также обеспечено высокопроизводительное и устойчивое функционирование графического интерфейса. При запуске программы происходит процесс сбора информации с последующим распределением их в отдельные части приложения.

За создание окна отвечает системная функция «CreateWindowEx()», в ней же указывается идентификатор окна, положение, размер и параметры, которыми будет обладать создаваемое окно. В контексте данной функции осуществляется не только формальное выделение атрибутов, но и строгая установка параметров, определяющих визуальные и функциональные аспекты окна. На рисунке 12 демонстрируется реализация пользовательской функции, ответственной за обработку различных изменений в окне. Эта функция активно взаимодействует с пользовательским интерфейсом, обеспечивая обработку событий, таких как нажатия на кнопки внутри окна, навигацию по строке меню и обновление данных в поле окна.


```

;=====
Procedure WindowCallback(hWnd, Msg, wParam, lParam)

Select Msg
  Case #WM_CREATE
    MainWndAddMenus(hWnd)
  Case #WM_COMMAND
    Select wParam
      Case #OnMenuCPU
        CPUMenu(hWnd)
      Case #OnMenuMotherboard
        MainboardMenu(hWnd)
      Case #OnMenuRAM
        RAMMenu(hWnd)
    EndSelect
  Case #WM_CLOSE
    DestroyWindow(hWnd)
  Case #WM_DESTROY
    PostQuitMessage(0) : Result = 0
  Default
    Result = DefWindowProc_(hWnd, Msg, wParam, lParam)
EndSelect

ProcedureReturn Result
EndProcedure
;=====

```

Рисунок 12 – «Функция, отвечающая за активность в приложении»

В процессе активного выполнения приложения окно остается в состоянии постоянного циклического обновления, что представляет собой механизм, обеспечивающий непрерывное отслеживание пользовательских действий. Этот бесконечный цикл гарантирует, что любое изменение, зафиксированное в виде, например, нажатия кнопки в меню, будет обнаружено и обработано, с последующим выполнением соответствующего ему алгоритма. В случае нажатия кнопки меню «CPU», функция «WindowCallback()» зафиксирует нажатие в области окна, и предпримет ответные меры, в данном случае алгоритм кода вызовет функцию, изображенную на рисунке 13а, которая в свою очередь начнет отрисовывать поля в окне и заполнит его данными.

```

Procedure CPUMenu(hWnd)
;===== CPU =====
CreateWindow_("static", "", #WS_VISIBLE | #WS_CHILD, 0, 0, 500, 400, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Processor", #WS_VISIBLE | #WS_CHILD, 5, 5, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Specification", #WS_VISIBLE | #WS_CHILD, 35, 35, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + CPUName(), #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 35, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Manufacturer", #WS_VISIBLE | #WS_CHILD, 34, 65, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + Manufacturer, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 65, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Socket", #WS_VISIBLE | #WS_CHILD, 75, 95, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + Socket, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 95, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Cores", #WS_VISIBLE | #WS_CHILD, 80, 125, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + Cores, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 125, 35, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Threads", #WS_VISIBLE | #WS_CHILD, 250, 125, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + Threads, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 310, 125, 35, 20, hWnd, #Null, #Null, #Null)

EndProcedure

```

Рисунок 13а – «Функция заполнения окна процессора»

То же самое относится к функции отрисовки материнской платы и ОЗУ, рисунок 13б и 13в соответственно.

```

Procedure MainboardMenu(hWnd)
;==== Mainboard ====
CreateWindow_("static", "", #WS_VISIBLE | #WS_CHILD, 0, 0, 500, 400, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Motherboard", #WS_VISIBLE | #WS_CHILD, 5, 5, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Manufacturer", #WS_VISIBLE | #WS_CHILD, 35, 35, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + MB_Manufacturer, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 35, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Model", #WS_VISIBLE | #WS_CHILD, 75, 65, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + MB_Model, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 65, 350, 20, hWnd, #Null, #Null, #Null)
;==== BIOS ====
CreateWindow_("static", "BIOS", #WS_VISIBLE | #WS_CHILD, 5, 125, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Brand", #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 75, 155, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + BIOSBrand, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 155, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Version", #WS_VISIBLE | #WS_CHILD, 75, 185, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + BIOSVersion, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 185, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Date", #WS_VISIBLE | #WS_CHILD, 75, 215, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + BIOSDate, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 215, 350, 20, hWnd, #Null, #Null, #Null)
EndProcedure

```

Рисунок 13б – «Функция заполнения окна материнской платы»

```

Procedure RAMMenu(hWnd)
;==== RAM ====
CreateWindow_("static", "", #WS_VISIBLE | #WS_CHILD, 0, 0, 500, 400, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "RAM", #WS_VISIBLE | #WS_CHILD, 5, 5, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Manufacturer", #WS_VISIBLE | #WS_CHILD, 35, 35, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + RAMManufacturer, #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 35, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Speed", #WS_VISIBLE | #WS_CHILD, 80, 65, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + RAMSpeed + " MHz", #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 65, 350, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", "Capacity", #WS_VISIBLE | #WS_CHILD, 65, 95, 100, 20, hWnd, #Null, #Null, #Null)
CreateWindow_("static", " " + RAMMemoryALL + " MB", #WS_VISIBLE | #WS_CHILD | #WS_BORDER, 130, 95, 350, 20, hWnd, #Null, #Null, #Null)
EndProcedure

```

Рисунок 13в – «Функция заполнения окна оперативной памяти»

Графический пользовательский интерфейс реализован в виде простого, но интуитивно понятного системного окна, предоставляющего удобный доступ к функционалу программы. Это окно содержит заранее созданное меню, включающее в себя три различных кнопки для перехода между разными вкладками. Каждая кнопка меню инициирует обновление экрана, отображающего информацию о выбранных комплектующих, указанных в названиях соответствующих вкладок. Рисунки 14а, 14б и 14в служат иллюстрациями для наглядного представления отображения выбранных комплектующих через графический пользовательский интерфейс в виде окон.

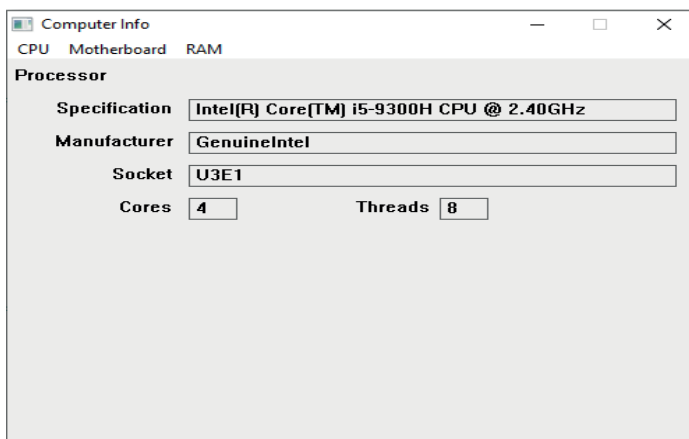


Рисунок 14а – «Окно с информацией о процессоре»

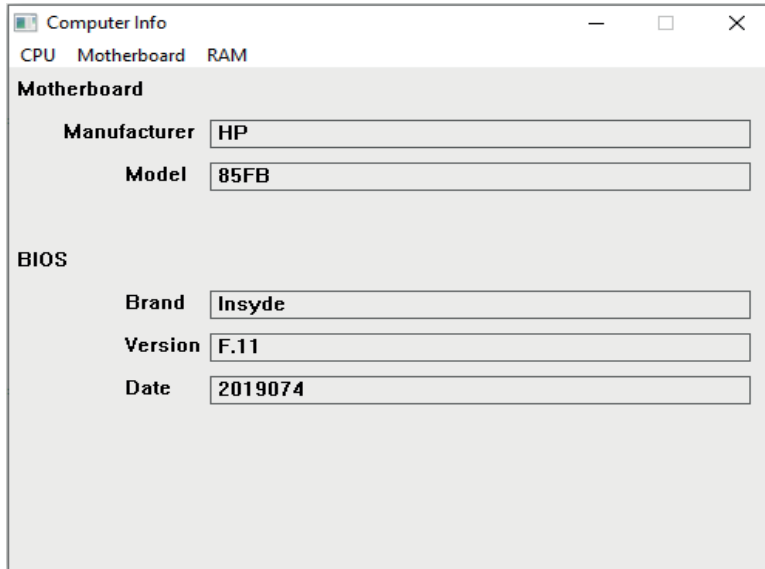


Рисунок 14б – «Окно с информацией о материнской плате»

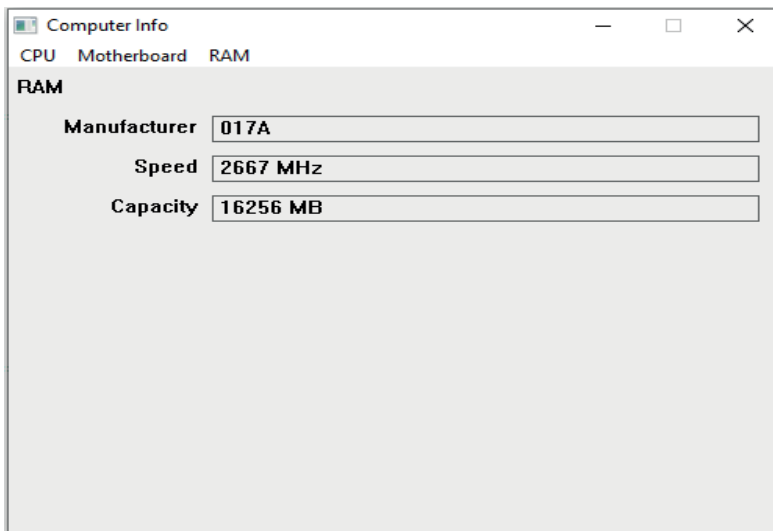


Рисунок 14в – «Окно с информацией о ОЗУ»

Заключение

Реализация методики проведения испытаний и нагрузочного тестирования компьютеров, с фокусом на создание графического пользовательского интерфейса и системы мониторинга, представляет собой важный шаг в области обеспечения надежности и эффективности вычислительных систем. В ходе этого исследования были преодолены технические вызовы и достигнуты ключевые цели, сделав акцент на следующих ключевых моментах:



Автоматизация и удобство использования. Разработанный графический пользовательский интерфейс значительно упрощает процесс проведения тестов и анализа результатов. Пользователь может легко взаимодействовать с системой, мониторить информацию о компьютерных комплектующих, не обладая глубокими техническими знаниями.

Точность и актуальность данных. Модуль определения комплектующих компьютера обеспечивает точный и актуальный сбор данных. Любое изменение аппаратной комплектации фиксируется и не остается незамеченными.

Гибкость и расширяемость. Реализованные решения обладают гибкостью и расширяемостью, что позволяет легко внедрять изменения, добавлять новые функциональности и адаптироваться к разнообразным требованиям тестирования.

Дальнейшее развитие. Эта работа является отправной точкой для дальнейшего развития. Дальнейшие шаги могут включать в себя оптимизацию производительности, расширение функционала GUI, добавление новых метрик для мониторинга и усовершенствование методов обработки данных.

ЛИТЕРАТУРА

Afef J.M., Moez K., Mohamed J. (2015). A comparative evaluation of state-of-the-art load and stress testing approaches.

Рэндал Э. Брайант, Дэвид Р. О'Халларон. (2015). Компьютерные системы. Архитектура и программирование.

Лазарева Н.Б., Горбачев К.А. (2020). Системы мониторинга оборудования.

Туровец Н.О., Алефиренко В.М. (2022). Методы тестирования интегрированных информационных систем.

Уткин Г.С., Башарин А.П. (2009). Особенности построения модели нагрузочного тестирования.

Марапулец Ю.В. (2019). Системное программирование в WINAPI.

Кораблин Ю.П., Косакян М.Л., Кучугуров И.В. (2013). Вопросы взаимодействия и синхронизации распределенных программ на C++ с использованием Win API.

REFERENCES

Afef J.M., Moez K., Mohamed J. (2015). A comparative evaluation of state-of-the-art load and stress testing approaches.

Randal E. Bryant and David R. (2015). O'Hallaron - Computer Systems: A Programmer's Perspective, — 3/E.

Lazareva N.B., Gorbachev K.A. (2020). Sistemy monitoringa oborudovaniya.

Turovec N.O., Alefirenko V.M. (2022). Metody testirovaniya integrirovannyh informacionnyh sistem.

Utkin G.S., Basharin A.P. (2009). Osobennosti postroeniya modeli nagruzochного testirovaniya.

Marapulec Yu.V. (2019). Sistemnoe programmirovaniye v WINAPI.

Korablin Yu.P., Kosakyan M.L., Kuchugurov I.V. (2013). Voprosy vzaimodejstviya i sinhronizacii raspredelennyh programm na C++ s ispol'zovaniem Win API.



**ХАЛЫҚАРАЛЫҚ АҚПАРАТТЫҚ ЖӘНЕ
КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР ЖУРНАЛЫ**

**МЕЖДУНАРОДНЫЙ ЖУРНАЛ ИНФОРМАЦИОННЫХ И
КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ**

**INTERNATIONAL JOURNAL OF INFORMATION AND
COMMUNICATION TECHNOLOGIES**

Правила оформления статьи для публикации в журнале на сайте:

<https://journal.iitu.edu.kz>

ISSN 2708–2032 (print)

ISSN 2708–2040 (online)

Собственник: АО «Международный университет информационных технологий» (Казахстан, Алматы)

ОТВЕТСТВЕННЫЙ РЕДАКТОР

Раушан Жалиқызы

КОМПЬЮТЕРНАЯ ВЕРСТКА

Жадыранова Гульнур Даутбековна

Подписано в печать 15.03.2024.

Формат 60x881/8. Бумага офсетная. Печать - ризограф. 9,0 п.л. Тираж 100
050040 г. Алматы, ул. Манаса 34/1, каб. 709, тел: +7 (727) 244-51-09).