

Крайник Я. М.,канд. техн. наук, в.о. ст. викладач
кафедри комп'ютерної інженерії,
e-mail -----**Гайван В. Ю.,**студент 4 курсу
спеціальності комп'ютерна інженерія,
e-mail -----**Килимович О. О.,**студент 4 курсу
спеціальності комп'ютерна інженерія,
ЧНУ ім. Петра Могили,
м. Миколаїв, Україна,
e-mail -----

РОЗРОБКА LOW-COST МУЛЬТИМЕДІЙНОГО НАВЧАЛЬНОГО КЛАСУ

У роботі запропонований підхід до розробки мультимедійного навчального класу університету, який дозволить його реалізацію на пристроях з низькими обчислювальними потужностями, а, значить, зменшити вартість створення такого класу. Розроблена модель взаємодії між клієнтом та сервером, яка дозволяє зменшити використання ресурсів пам'яті пристрою за рахунок того, що наявність операційної системи на пристрої не є необхідною. Реалізовано прототип, який показав, що такий підхід для побудови може використовуватись при створенні нових навчальних класів університету.

Ключові слова: мультимедійний клас; сервер; клієнт; API.

Постановка проблеми та аналіз основних досліджень. Безперечним є те, що комп'ютерні технології увійшли у всі сфери суспільного життя та галузі виробництва. Тому використання їх потенціалу і людських можливостей в менеджменті освітньої сфери може не лише сприяти реалізації нових технологій навчання, а й істотно поліпшити якість знань студентів. Спосіб, який дозволяє здійснити це, – комплексна комп'ютеризація учебного процесу, а саме розробка комп'ютерного класу.

Розроблення концепції комп'ютеризації розпочалося у 80-х роках минулого сторіччя. Швидкий розвиток технічних і програмних можливостей персональних комп'ютерів, розповсюдження інформаційно-комунікаційних і креативних технологій створюють реальні можливості для їх використання в системі освіти з метою розвитку творчого потенціалу людини в процесі навчання та забезпечення наступності навчання між різними ланками навчально-виховних закладів освіти.

Розвиток всіх наук характеризується не тільки новаціями в галузі методів, способів та організаційних форм навчання. На різних етапах навчання з технічним розвитком відбувалося інтенсивне впровадження у навчальний процес різноманітних засобів наочності, технічних засобів. Все це здійснювалося з однією метою: підвищення ефективності навчального процесу в цілому й інновацій, які запроваджуються.

Необхідно зауважити, що безпосередньо для навчального процесу було розроблено небагато технічних пристроїв. Серед них можна назвати лише графопроєктор і лінгафонний пристрій. Всі інші технічні засоби запозичені з побуту: програвач, магнітофон, телевізор, відеоманітофон, комп'ютер. Щоразу, під час появи нового технічного пристрою, педагоги розмірковують, як використати його для розв'язання дидактичних завдань.

Таким чином, аналізуючи природні якості засобів навчання, викладачі визначають їх дидактичні властивості та функції в навчальному процесі. Учні пізнають навколишній світ за допомогою всіх органів почуттів. Проте, основними каналами отримання інформації є зорові аналізатори. Тому не випадково, що 90 відсотків всієї інформації учні отримують за допомогою зору, а 10 % – за допомогою слуху. «Краще один раз побачити, ніж 100 разів почути» – так говорить досить актуальне прислів'я.

Саме тому нами було вирішено розробити систему для представлення навчального контенту для низько вартісного навчального мультимедіа-класу.

Даний проект не є новизною, схожі речі уже було створені до нас. Але унікальність нашого проекту полягає в тому, щоб суттєво скоротити витрати на створення мультимедійного класу, обладнання для якого є звичай не дешевим. Не зважаючи на те, що вартість буде істотно меншою, ніж у конкурентів,

функціонал в основному нічим не поступається аналогам.

Постановка завдання. Основними цілями цього проекту є розробка системи для низьковартісного навчального мультимедіа-класу, що здійснювалась за двома напрямками:

- представлення навчального контенту;
- перевірка знань.

Для цього необхідно розробити модель взаємодії клієнт-сервер, яка дозволить зменшити використання ресурсів пам'яті клієнтського пристрою на 10–15 %.

Об'єктом роботи є мультимедійний клас. Предметом роботи є використання мультимедійного класу для проведення викладачем пар з використанням презентаційних матеріалів та перевіркою знань студентів.

Основна частина. Для передачі інформації від сервера до клієнтів ми використали lightweight IP. Це широко використовуваний TCP/IP [1] стек з відкритим вихідним кодом, призначений для вбудованих систем. LwIP був спочатку розроблений Адамом Дункелем в Шведському інституті комп'ютерних наук і в даний час розробляється і підтримується всесвітньою мережею розробників на чолі з Kiran Mansley.

У центрі уваги реалізації LwIP TCP/IP [6], що зменшує використання оперативної пам'яті, в той же час маючи повномасштабну TCP. Ця конструкція LwIP підходить для використання у вбудованих системах з десятками кілобайт вільної оперативної пам'яті і файлів розміром близько 40 кб коду ROM.

Очевидно, що підтримку всього різноманіття пакетів і стандартів потрібно покласти на мережевий стек. Саме тому було прийнято рішення використовувати популярний стек для вбудованих систем LwIP останньої (на поточний момент) версії.

Протоколи транспортного рівня, такі в ієрархії слідує за IP, використовуються для передачі даних між прикладними процесами, що реалізуються в мережевих вузлах. Пакет даних, що надійшов від одного комп'ютера іншому через Інтернет, повинен бути переданий процесу-обробнику, і саме по конкретному призначенню. Транспортний рівень приймає на себе відповідальність за це. На цьому рівні два основні протоколи: TCP і UDP.

TCP – транспортний протокол передачі даних в мережах TCP / IP, попередньо встановлює з'єднання з мережею.

UDP – транспортний протокол, який передає повідомлення-датаграми без необхідності установки з'єднання в IP-мережі.

Різниця між протоколами TCP і UDP - в так званій "гарантії доставки". TCP вимагає відгуку від клієнта, якому доставлений пакет даних, підтвердження доставки, і для цього йому необхідно встановлене заздалегідь з'єднання. Протокол TCP вважається надійним, тоді як UDP тоді отримав навіть іменування «протокол ненадійних датаграм». TCP виключає втрати даних, дублювання і перемішування пакетів, затримки. UDP все це допускає, і з'єднання для роботи йому не потрібно. Процеси, яким дані передаються по UDP, повинні обходитися отриманим, навіть і з втратами. TCP контролює завантаженість з'єднання, UDP не

контролює нічого, крім цілісності отриманих датаграм.

З іншого боку, завдяки такій невибірковості і безконтрольності, UDP доставляє пакети даних (дейтаграми) набагато швидше, тому для додатків, які розраховані на широку пропускну здатність і швидкий обмін, UDP можна вважати оптимальним протоколом. До таких належать мережеві і браузерні ігри, а також програми перегляду потокового відео і додатки для відеозв'язку (або голосовий): від втрати пакету, повної або часткової, нічого не змінюється, повторювати запит не обов'язково, зате завантаження відбувається набагато швидше. Протокол TCP, як більш надійний, з успіхом застосовується навіть в поштових програмах, дозволяючи контролювати не тільки трафік, але і довжину повідомлення та швидкість обміну трафіком.

Відмінність TCP від UDP

– TCP гарантує доставку пакетів даних в незмінній вигляді, послідовності і без втрат, UDP нічого не гарантує.

– TCP вимагає заздалегідь встановленого з'єднання, UDP з'єднання не вимагає.

– UDP забезпечує більш високу швидкість передачі даних.

– TCP надійніше і здійснює контроль над процесом обміну даними.

– UDP краще для програм, що відтворюють потокове відео, відеофонії і телефонії, мережевих ігор.

Отже, на підставі наведених аргументів, було вирішено використати TCP протокол передачі даних, адже для нас якість передачі даних грає набагато важливішу роль, ніж швидкість.

Наступним етапом був вибір API. API – інтерфейс прикладного програмування набір стандартних програмних переривань, викликів процедур (функцій, методів) і форматів даних, які повинні використовувати прикладні програми для запиту і отримання від операційної системи, телекомунікаційного протоколу або програмного інтерфейсу (механізму) і т. п. пов'язаного з ними обслуговування. API визначає на рівні початкового тексту деякий рівень абстракції, який дозволяє переносити вихідні тексти програм на комп'ютери з відмінними процесорами, де вони після перекompіляції зможуть відразу виконуватися.

Бібліотека для роботи з мережевими інтерфейсами lwIP [3] надає можливість реалізовувати програмні застосування за допомогою трьох різних API – наборів (англ. Application Programming Interface):

1. Raw API.
2. Netconn API.
3. Socket API.

Кожна з цих альтернатив має свої переваги та недоліки, які необхідно враховувати відповідно до функціональних потреб системи.

Raw API [5] дозволяє отримати доступ до керування мережевим інтерфейсом на найнижчому рівні. За рахунок цього можливо досягти найбільшої пропускну здатності та зменшити використання пам'яті, яка необхідна для роботи з керуючими структурами бібліотеки. Тим не менш, доступ до апаратних ресурсів на низькому рівні зумовлює ускладнення програмної реалізації. Використання Raw API передбачає

необхідність коректної реалізації скінченого автомату для роботи з подіями (прийом, отримання, підключення) мережевого інтерфейсу.

Інші варіанти API – Netconn та Socket – є простішими з точки зору програмної моделі. Socket API є спрощеною версією широко розповсюдженого інтерфейсу сокетів. Завдяки цьому безпосередньо робота з прийомом/передачею даних є набагато простішою та звичною, ніж для Raw API. Netconn API є нативним інтерфейсом характерним саме для бібліотеки lwIP [2]. Даний набір програмних засобів оптимізований (менше використання пам'яті, вища пропускна здатність) у порівнянні з сокетами. Дана частина бібліотеки підтримує більшість можливостей стандартних сокетів. Однак, використання даного програмного інструменту (це стосується і Netconn) вимагає наявності операційної системи, що означає збільшення вимог до розміру пам'яті цільового пристрою, а також додавання ще одного програмного компоненту.

Вибір API для реалізації залежить від функціональних вимог до системи. Оскільки система мультимедійного класу потребуватиме забезпечення високої пропускної здатності, то обраним варіантом у даному випадку є Raw API.

На початковому етапі система має працювати в 2 режимах:

1. Тестування.
2. Демонстрація презентацій, наочних матеріалів тощо.

На даному етапі для управління процесом необхідно реалізувати сервер, який розміщуються на робочому комп'ютері (комп'ютер викладача) та відповідальний за взаємодію з клієнтами.

Програма-сервер на даному етапі повинна реалізувати наступний функціонал:

1. Підтримувати підключення кількох клієнтів (на кожного клієнта відводиться окремий потік) [4]. Підключення використовує протокол TCP. Клієнт передає серверу ідентифікатор одразу після того, як йому

вдалось успішно підключитись. Ідентифікатор – номер логіну студента. Даний ідентифікатор сервер використовує для того, щоб розрізнити клієнтів.

2. У режимі демонстрації презентації одні й ті самі дані (1 зображення зі списку передається усім клієнтам). Реалізації можуть бути різними: вказується список файлів-зображень, які відправляються по натисненню кнопки Next або Previous, або вказується лише папка, а програма на основі розширення визначає, які файли доступні для передачі. При передачі зображення клієнтам необхідно також виводити його на форму для того, щоб викладач мав змогу бачити те, що відображається для студентів.

3. У режимі організації опитування необхідно передавати усім підключеним клієнтам запитання відповідно до того, як відбувається проходження тесту. Програма має перевіряти відповіді по закінченню тесту, видавати результати щодо успішності та передавати відповідні результати клієнтам. На початку усім підключеним клієнтам передається одне й те саме питання, прогрес відносно наступних питань може бути різним, тому обробка даного режиму обов'язково має бути реалізована в окремому потоці.

На рисунку 1 зображена діаграма варіантів використання створеного програмного забезпечення. Програмним забезпеченням можуть користуватись дві групи користувачів: викладачі та студенти. За допомогою створеного програмного забезпечення викладач може переслати зображення студентам, або переслати файл з тестами, для перевірки знань студентів. Для пересилання зображення викладач повинен обрати зображення з своєї файлової системи. Для тестування студентів викладачу необхідно з своєї файлової системи обрати файл з питаннями тестів та файл з вірними відповідями для перевірки. Група студентів має можливість переглянути зображення, яке переслав викладач та пройти тестування. Протокол взаємодії при передачі даних клієнту є наступним (рис. 2).

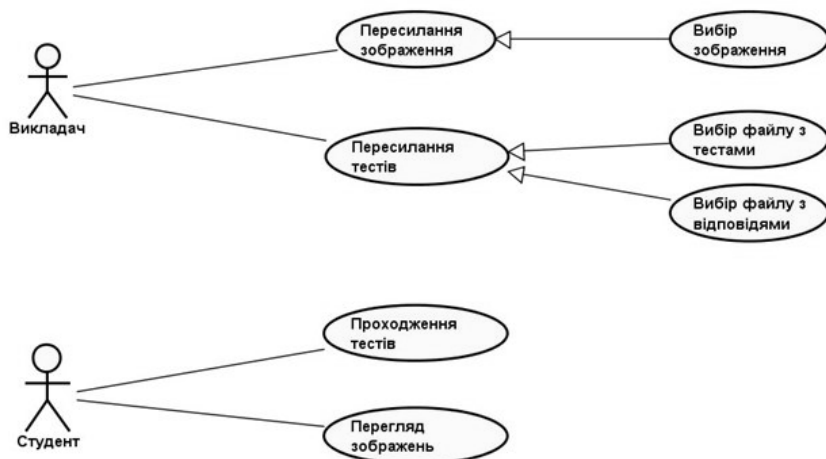


Рис. 1. Діаграма варіантів використання програмного забезпечення

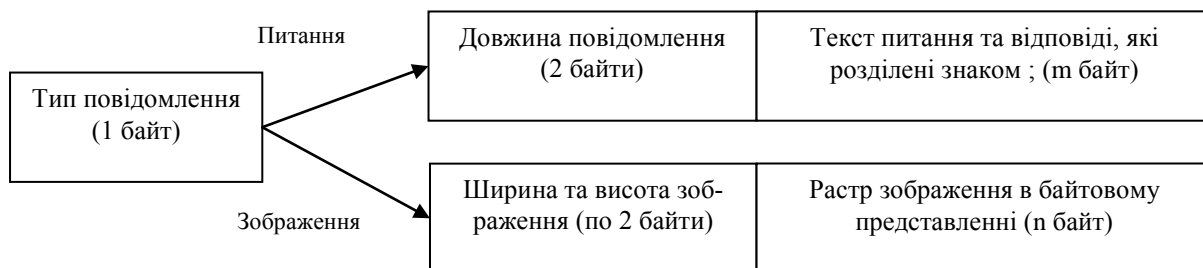


Рис. 2. Протокол передачі повідомлень від серверу до клієнтів

У режимі демонстрації ніяких даних від клієнтів до серверу не передається. У режимі тестування клієнти передають номер обраного варіанту відповіді (1 байт).

Реалізація працює на основі віконного інтерфейсу та у подальшому буде змінюватись відповідно до прогресу в реалізації.

Для забезпечення нормальної функціональності даного проекту необхідно правильно підключити необхідні частини мережі. В першу чергу потрібно запустити додаток на комп'ютері, який буде сервером. З допомогою додатку можна обрати режим роботи: тести або демонстрація презентації. Необхідні дані можна обрати серед наявних на комп'ютері-сервері. За

допомогою Ethernet-кабелю (конектори RJ-45) потрібно під'єднати switch. На цьому етапі опис зі сторони серверу закінчено.

Клієнтська частина представляє собою довільну кількість плат (STM32F7), із заздалегідь завантаженою на них прошивкою (але не більшою, ніж це дозволяє switch). Кожна плата під'єднується за допомогою кабелю Ethernet з роз'ємом RJ-45 до switch'a. Живиться плата за допомогою гнізда mini-USB. Реалізувати це можна за допомогою кабелю mini-USB – USB, під'єднавши його до порту USB. Живлення заданої кількості плат можна забезпечити через USB-hub, попередньо під'єднавши його до комп'ютера (наприклад комп'ютера-сервера).

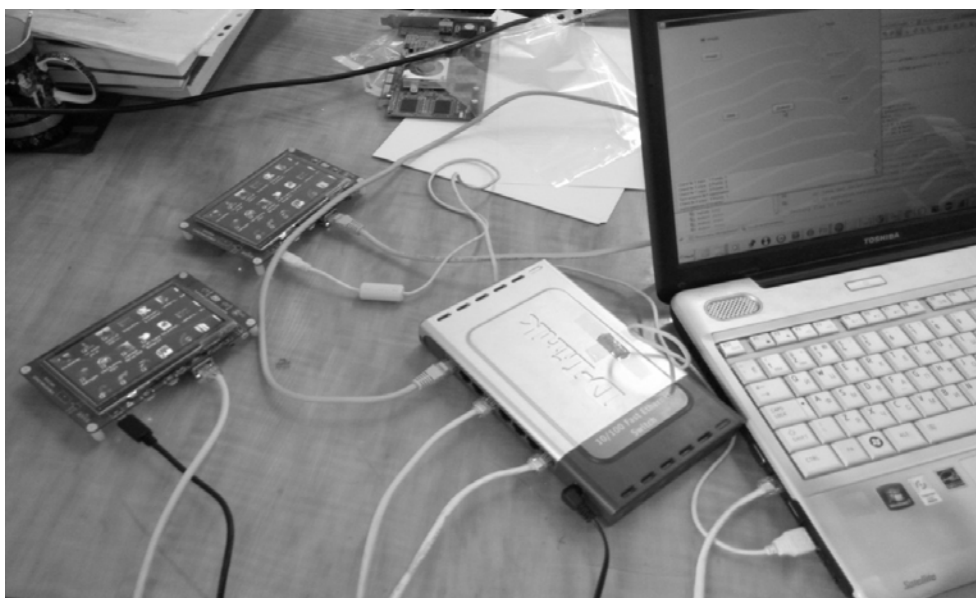


Рис. 3. Демонстрація роботи у режимі презентації

При запуску програми сервер очікує на підключення клієнтів. При кожному підключенні нового клієнта створюється новий потік, надалі в якому він і працює. У разі підключення нового клієнта йому передаються дані у вигляді зображення або тестів, які безпосередньо обрані в серверній частині. Одразу після закінчення тестів їх результат виводиться на екран.

На рисунку 4 зображена діаграма класів створеного програмного забезпечення. Програмне забезпечення містить три основні класи. Клас mydiplomUI є го-

ловним та містить в собі компоненти класу JFrame (поле для виводу, кнопки, форми для вибору файлів з завданнями). Цей клас ініціалізує об'єкти, які знаходяться на панелі та створює серверний потік. У класі ServerThread описується робота сервера та створюється потік класу ClientServicThread. Даний клас містить один метод, який запускає даний потік та успадковується від інтерфейсу Runnable. У класі ClientServicThread описується робота клієнтської частини. Даний клас успадковується від класу Thread.

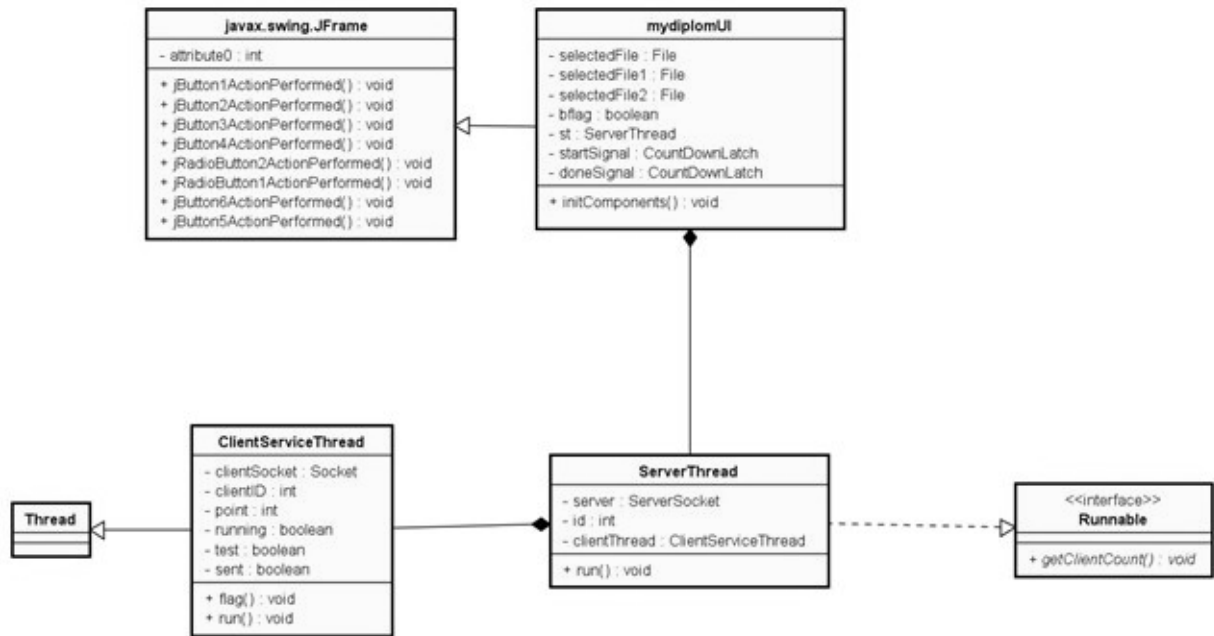


Рис. 4. Діаграма класів програмного забезпечення

Для тестування нашої мережі було прийнято рішення використати Wireshark. Wireshark (раніше звався Ethereal) – програма для аналізу мережевих пакетів Ethernet і інших мереж (сніфер) з вільним вихідним кодом. Має графічний інтерфейс користувача, програма розпізнає структуру найрізноманітніших мережевих протоколів, і тому дозволяє розібрати мережевий пакет, відображаючи значення кожного поля

протоколу будь-якого рівня. Оскільки для захоплення пакетів використовується rpsar, існує можливість захоплення даних тільки з тих мереж, які підтримують цю бібліотеку. На рисунку 5 зображено результат тестування проекту за допомогою даної програми. Можна переглянути тип переданого повідомлення та його довжину, а також його вміст.

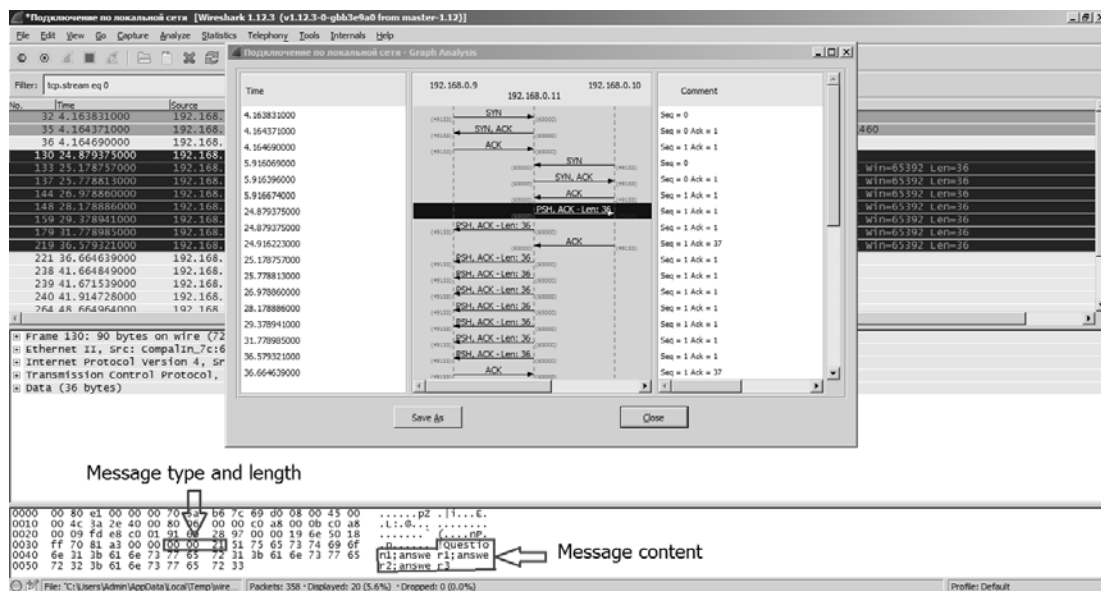


Рис. 5. Тестування проекту за допомогою Wireshark

Висновки

За результатами виконання роботи вдалося досягти зазначеної мети, а саме створити прототип низько вартісного мультимедійного класу, робота якого здійснювалась за двома напрямками:

- представлення навчального контенту;
- перевірка знань.

Цей клас реалізовано на основі моделі взаємодії клієнт-сервер, що дозволило зменшити використання ресурсів пам'яті клієнтського пристрою на 10 % за рахунок відмови від використання операційної системи.

Було розроблено програмне забезпечення для легкого управління мультимедійного класу викладачем. Завдяки розробленому проекту можна значно спрос-

тити та полегшити процес навчання, адже кожен студент індивідуально матиме змогу ознайомитись з навчальним матеріалом (особливо це корисно для студентів з вадами зору, які не можуть бачити на досить значні відстані представлений матеріал), а також в залежності від швидкості проходження тесту виводити нові запитання (кожен студент працює з різною

швидкістю, тобто на кожне питання може піти різна кількість часу). На підставі наданих відповідей кожен студент моментально отримує результати тестування з виведенням їх на плату. Цей проект є досить актуальним у наш час, адже це може позитивно вплинути на якість навчання студентів.

ЛІТЕРАТУРА

1. Developing applications on STM32Cube with LwIP TCP/IP stack [Електронний ресурс]. – Режим доступу : http://www.st.com/content/ccc/resource/technical/document/user_manual/65/e8/20/db/16/36/45/f7/DM00103685.pdf/files/DM00103685.pdf/jcr:content/translations/en.DM00103685.pdf. – Назва з екрану.
2. Dunkels, A. Design and Implementation of the lw IP TCP/IP Stack [Текст] / A. Dunkels. – Swedish Institute of Computer Science, 2001. – 11с.
3. lwIP API [Електронний ресурс]. – Режим доступу : http://www.nongnu.org/lwip/2_0_0/raw_api.html. – Назва з екрану.
4. Multithreaded Server in Java [Електронний ресурс]. – Режим доступу : <http://tutorials.jenkov.com/java-multithreaded-servers/multithreaded-server.html>. – Назва з екрану.
5. Raw/TCP [Електронний ресурс]. – Режим доступу: <http://lwip.wikia.com/wiki/Raw/TCP>. – Назва з екрану.
6. Установка TCP-соединения в стеке lwIP [Електронний ресурс]. – Режим доступу : http://cathechysis.ru/lwip_tcp_connect/. – Назва з екрану.

**Я. М. Крайнык,
В. Ю. Гайван,
О. О. Килимович,**

Черноморский национальный университет
им. Петра Могилы,
г. Николаев, Украина

РАЗРАБОТКА LOW-COST МУЛЬТИМЕДИЙНОГО УЧЕБНОГО КЛАССА

В работе предложен подход к разработке мультимедийного учебного класса университета, который позволит выполнить его реализацию на устройствах с низкими вычислительными возможностями, а, значит, уменьшить стоимость такого класса. Разработана модель взаимодействия между клиентом и сервером, которая позволяет уменьшить использование ресурсов памяти устройства за счет того, что наличие операционной системы не является необходимым. Реализован прототип устройства, который показал, что предложенный подход может реализовываться для создания учебных классов университета.

Ключевые слова: мультимедийный класс; сервер; клиент; API.

**Y. M. Krainyk,
V. Yu. Haivan,
O. O. Kylymovych,**

Petro Mohyla Black Sea National University,
Mykolayiv, Ukraine

DEVELOPMENT OF LOW-COST MULTIMEDIA LEARNING CLASS

The new approach to the development of multimedia learning class in university is proposed. It allows implementing class based on devices with low computation abilities, which means decreasing total cost of the class. Developed model of interconnection between clients and servers that makes it possible to decrease resource consumption. This result has been achieved due to the fact that model makes unnecessary usage of operating system. Implemented prototype proves that proposed approach can be utilized for learning class design in university.

Key words: multimedia class; server; client; API.

Рецензенти: д. т. н., проф. *М. П. Мусієнко*;
к. т. н., доц. *І. М. Журавська*.

© Крайник Я. М., Гайван В. Ю., Килимович О. О., 2016

Дата надходження статті до редколегії 23.11.16