

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение «Томский
государственный университет систем управления и радиоэлектроники»
(ТУСУР)

УТВЕРЖДАЮ

Заведующий кафедрой
«Управление инновациями»

_____ А.Ф. Уваров

«___» _____ 2012 г.

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ,
ЛАБОРАТОРНЫМ РАБОТАМ И САМОСТОЯТЕЛЬНОЙ РАБОТЕ**

по дисциплине

«АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ»

Составлены кафедрой «Управление инновациями»

Для студентов, обучающихся по направлению подготовки 222000.68 «Инноватика»

Магистерская программа «Управление инновациями в электронной технике»

Форма обучения – очная

Составитель:

ассистент кафедры УИ

_____ И.И. Надреев

«___» _____ 2012 г.

Томск 2012 г.

ОГЛАВЛЕНИЕ

Введение.....	3
Практическое занятие №1. «Знакомство с операционной системой Linux на примере дистрибутива Ubuntu. Запуск программ. Работа с файловой системой. Сервисы.»	4
Практическое занятие №2. «Исследование сетевых потоков с помощью программы Wireshark. Использование программы tcpdump.»	6
Практическое занятие №3. «Знакомство с инструментарием создания программного обеспечения в операционной системе Linux. Компиляторы. Средства сборки. Среды разработки. Средства поддержки разработки.»	8
Практическое занятие №4. «Знакомство с архитектурами параллельных вычислений nVidia CUDA, AMD FireStream, OpenCL. Инструментарий разработки».....	10
Лабораторная работа №1. «Исследование аппаратной архитектуры компьютера с использованием возможностей командной оболочки Ubuntu Linux.»	12
Лабораторная работа №2. «Создание и исследование простого клиент-серверного приложения».....	14
Лабораторная работа №3. «Исследование простого приложения на базе архитектуры CUDA».....	15
Самостоятельная работа №1. «Разработка проекта по представленной UML-диаграмме».....	16
Самостоятельная работа №2. «Применение технологии параллельных вычислений для задач обработки изображений».....	18
ЛИТЕРАТУРА	19

Введение

Изучение дисциплины «Архитектура вычислительных систем» (уровень дисциплины региональный (вузовский)) имеет основополагающее значение в специальной подготовке студентов по направлению «Инноватика».

Цель данного пособия состоит в выработке знаний и навыков в области организации современных компьютерных систем, процессов обработки информации и программирования.

Для полноценного понимания и усвоения материала необходимы знания информатики и основ информационных технологий.

Для углубленного изучения и освоения материала целесообразно применение различных форм самопроверки знаний студентов: тесты, задачи, упражнения. Они могут быть использованы при проведении практических занятий в университете, выполнении курсовых, контрольных и аудиторных работ, а также при самостоятельном изучении данных дисциплин.

Одним из наиболее интенсивных способов изучения дисциплины является самостоятельное решение практических задач. При этом вырабатываются навыки оценки сложности поставленной задачи, сроков ее реализации и требующихся для этого программно-аппаратных средств.

Предлагаемые задания позволят глубже освоить теоретические и практические вопросы устройства вычислительных систем, понять принципы работы программного обеспечения, узнать о роли, организации и функциональности операционных систем, а также научиться применять полученные знания при проектировании и разработке компьютерных систем и сетей на их основе.

Практическое занятие №1. «Знакомство с операционной системой Linux на примере дистрибутива Ubuntu. Запуск программ. Работа с файловой системой. Сервисы.»

Цель занятия:

Получить представление о роли, структуре и функциональности операционной системы вообще и операционной системы на базе ядра GNU Linux в частности, на примере дистрибутива Ubuntu Linux 12.04. Познакомиться с основами работы в операционной системе Ubuntu Linux. Узнать о назначении и устройстве файловой системы. Получить представление о сервисах операционной системы.

Задание

Освоить работу в операционной системе Ubuntu Linux. Исследовать аппаратное устройство компьютера. Исследовать файловую систему. Установить программное обеспечение для разработки программ. Написать скрипт оболочки. Установить и запустить веб-сервер.

Порядок выполнения.

1. Получить на группу дистрибутив Ubuntu Linux 12.04 и установить его на компьютере.
2. Установить из репозитория программное обеспечение для разработки и последующих работ: GNU GCC, GNU GDB, Subversion client (SVN), Wireshark, tcpdump, Midnight Commander (mc), cmake, KDevelop.
3. Получить у преподавателя архив с исходным кодом пакета tstools. Распаковать его на компьютере и собрать бинарные утилиты. Установить их в операционную систему и протестировать работоспособность.
4. Исследовать аппаратное устройство компьютера с помощью средств операционной системы.

5. Написать скрипт оболочки, создающий файл с отчетом о аппаратном устройстве компьютера (характеристиках CPU, RAM, HDD)

6. Установить из репозитория веб сервер Apache. Запустить его сервис. Протестировать работу веб сервера. Изменить вариант загрузки сервиса.

Результатом практического занятия должно явиться практическое освоение методов работы в операционной системе на базе ядра GNU Linux.

Практическое занятие №2. «Исследование сетевых потоков с помощью программы Wireshark. Использование программы tcpdump.»

Цель занятия:

Познакомиться с принципами работы компьютерной сети. Изучить протоколы взаимодействия вычислительных систем на примере стека протоколов TCP/IP. Научиться исследовать сетевые потоки.

Задание

Изучить организацию компьютерной сети. Освоить работу с программой Wireshark. Научиться использовать утилиту tcpdump для анализа сетевых потоков.

Порядок выполнения.

1. Исследовать настройки и характеристики сетевых интерфейсов. Перезапустить их. Получить заново IP адрес по протоколу DHCP. Проверить работоспособность сетевого интерфейса и доступность сетевых устройств (используя команду ping).
2. Подключить (смонтировать) сетевую папку. Передать файл на сервер.
3. Получить файл по протоколу HTTP, используя программу wget.
4. Запустить программу Wireshark. Начать «захват» данных с использованием фильтра для определенного протокола, адреса и порта. Запустить «закачку» по протоколу HTTP, используя программу wget. Проанализировать полученный лог с использованием различной фильтрации. Сохранить HTTP обмен между клиентом и сервером.
5. Научить работать с программой tcpdump для захвата сетевого потока данных на примере захвата UDP multicast потока. Проанализировать захваченный поток с помощью программы Wireshark и пакета tstools.

Результатом практического занятия должно явиться практическое освоение принципов работы компьютерной с сети, сетевых протоколов, основ работы программным обеспечением анализа сетевых потоков.

Практическое занятие №3. «Знакомство с инструментарием создания программного обеспечения в операционной системе Linux. Компиляторы. Средства сборки. Среды разработки. Средства поддержки разработки.»

Цель занятия:

Изучить возможности и получить навыки работы с инструментарием создания программного обеспечения в операционной системе Linux.

Задание

Создать тестовый проект для сборки программы из исходных текстов. Найти ошибку в исходных текстах и собрать программу. Исправленную версию исходных текстов программы зафиксировать в системе контроля версий.

Порядок выполнения.

1. Написать исходный текст программы «Hello world». Собрать программу и проверить.
2. Загрузить исходные тексты тестового проекта из системы контроля версий SVN.
3. Создать makefile для сборки и установки тестового проекта.
4. Запустить сборку. Найти ошибку в исходных текстах и исправить ее. Собрать проект, установить его и проверить работоспособность.
5. Создать конфигурационный файл cmake для тестового проекта. Сгенерировать проект для среды разработки KDevelop. Запустить KDevelop и загрузить сгенерированный проект. Собрать проект в среде KDevelop. Запустить, собранный проект, под отладчиком GDB в среде KDevelop.
6. Зафиксировать в системе контроля версий исправленную версию исходных текстов программы.

Результатом практического занятия должно явиться приобретение навыков работы с системой контроля версий, инструментальными средствами сборки и отладки программных продуктов в среде операционной системы Linux.

Практическое занятие №4. «Знакомство с архитектурами параллельных вычислений nVidia CUDA, AMD FireStream, OpenCL. Инструментарий разработки».

Цель занятия:

Исследовать технологии параллельных вычислений на GPU. Исследовать особенности работы специализированных процессоров для параллельных вычислений.

Задание

Исследовать возможности создания программ с использованием технологий CUDA и OpenCL.

Порядок выполнения.

1. Определить с помощью средств операционной системы тип видеоадаптера, установленного на компьютере. Определить характеристики видеоадаптера с помощью утилит `xrpyinfo`, `glxinfo`, `xvinfo`.
2. Исследовать установленные драйверы видеоадаптера и определить поддержку технологий параллельных вычислений.
3. Установить Intel SDK for OpenCL Applications. Собрать и запустить тестовый пример.
4. Установить AMD Accelerated Parallel Processing (APP) SDK. Собрать и запустить тестовый пример.
5. Установить CUDA Toolkit и CUDA SDK. Собрать и запустить тестовый пример.

Результатом практического занятия должно явиться получение представления о технологиях параллельных вычислений, а также получения навыков создания программ, использующих данную технологию.

Лабораторная работа №1. «Исследование аппаратной архитектуры компьютера с использованием возможностей командной оболочки Ubuntu Linux.»

Цель занятия:

Исследовать аппаратную архитектуру компьютера. Получить опыт работы в среде операционной системы на ядре Linux вообще и навыки исследования аппаратной архитектуры в частности.

Задание

Исследовать аппаратную архитектуру компьютера используя команды оболочки операционной системы Linux.

Порядок выполнения.

1. Загрузить операционную систему. Запустить консоль оболочки. С помощью команд оболочки исследовать характеристики CPU компьютера.
2. Исследовать память и HDD.
3. Определить характеристики видеоадаптера с помощью утилит `xdrinfo`, `glxinfo`, `xvinfo`.
4. Получить список PCI-устройств и их характеристик.
5. Посмотреть модель материнской платы и получить краткую информацию о SCSI-устройствах.
5. Создать скрипт оболочки собирающий основную информацию о компьютере в текстовый файл отчета. Для этого использовать команды работы со строками (`grep`, `head` и т.д).

Результатом лабораторной работы должно явиться получение навыков работы в среде операционной системы типа Linux, создания скриптов

оболочки. Понимание основ взаимодействия аппаратного обеспечения компьютера и операционной системы.

Лабораторная работа №2. «Создание и исследование простого клиент-серверного приложения».

Цель занятия:

Получить навыки создания сетевого программного обеспечения. Получить знания в области сетевого взаимодействия вычислительных систем.

Задание

Разработать простое клиент-серверное приложения для передачи данных по протоколу TCP.

Порядок выполнения.

1. Создать, используя среду KDevelop, два проекта для реализации задания (Сервер и Клиент).
2. Разработать и проверить процедуры «прослушивания порта» сервером и подключения клиента к серверу.
3. Разработать передачу символьных данных сервером и прием их на стороне клиента.
4. Проверить работу клиент-серверного приложения, используя для анализа потока программу Wireshark.

Результатом практического занятия должно явиться получение навыков создания сетевого программного обеспечения, поиска и устранения ошибок, анализа сетевого трафика.

Лабораторная работа №3. «Исследование простого приложения на базе архитектуры CUDA»

Цель занятия:

Получить навыки реализации алгоритмов с использованием технологии параллельных вычислений на GPU.

Задание

Исследовать возможности технологий CUDA и OpenCL на примере реализации вычислительного алгоритма. Оценить преимущества использования технологий параллельных вычислений.

Порядок выполнения.

1. Проверить наличия на компьютере установленных дистрибутивов CUDA Toolkit, CUDA SDK, NVIDIA OpenCL SDK.
2. Реализовать с использованием CUDA трансформацию трех векторов вида $a \times b + c$.
3. Реализовать тестовый проект для много кратного запуска теста.
4. Реализовать альтернативный вариант трансформации трех векторов с использованием только CPU.
5. Реализовать вариант трансформации трех векторов с использованием архитектуры OpenCL.
6. Запустить тест и измерить скорость выполнения всех вариантов тестов. Сделать выводы о производительности каждого из вариантов реализации.

Результатом практического занятия должно явиться получение навыков разработки программного обеспечения с использованием технологий параллельных вычислений, а также осознание преимуществ данных технологий.

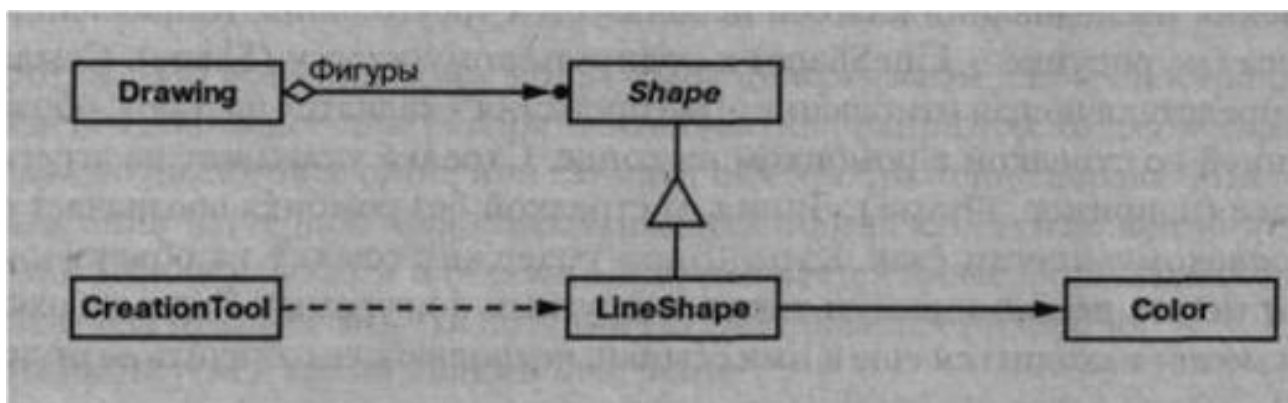
Самостоятельная работа №1. «Разработка проекта по представленной UML-диаграмме»

Цель занятия:

Получить навыки разработки программного обеспечения, опыт применения шаблонов проектирования и умение работать с UML диаграммами.

Задание

По представленной UML-диаграмме реализовать тестовый проект, используя среду разработки KDevelop и систему автоматизации сборки CMake.



Порядок выполнения.

1. Изучить UML-диаграмму и определить необходимые объекты и классы, которые потребуются для реализации проекта.
2. Для реализации проекта необходимо создать новый проект в среде KDevelop. Далее необходимо определить из каких модулей будет состоять проект, распределив классы и объекты по этим модулям.
3. После создания модулей можно приступать к реализации классов. При реализации графического вывода на экран, достаточно лишь того, чтобы каждый объект мог выводить информацию о себе в текстовом виде.
5. После реализации необходимо проанализировать исходный код, структуру проекта. Скомпилировать, запустить проект (проект содержит легко

обнаруживаемые ошибки компиляции и времени выполнения). Отчет о найденных ошибках включить в отчет.

Самостоятельная работа №2. «Применение технологии параллельных вычислений для задач обработки изображений».

Цель занятия:

Основываясь на знаниях, полученных в ходе лекций, познакомиться с возможностями технологий параллельных вычислений для задач обработки изображений. Изучить преимущества использования технологий параллельных вычислений.

Задание

1. Реализовать алгоритм медианной фильтрации изображений на языке C.
2. Реализовать алгоритм медианной фильтрации изображений с использованием выбранной технологии параллельных вычислений.
3. Измерить прирост производительности при использовании реализации алгоритма с применением технологии параллельных вычислений.

Порядок выполнения.

1. Изучить, используя источник [6], алгоритм медианной фильтрации изображений.
2. Реализовать алгоритм медианной фильтрации изображений на языке C в среде KDevelop, создав тестовый проект.
3. Реализовать алгоритм медианной фильтрации изображений с использованием выбранной технологии параллельных вычислений.
4. Реализовать набор тестов для оценки производительности различных реализаций алгоритма.
5. Протестировать реализации алгоритма. Сделать выводы и написать отчет с результатами работы.

ЛИТЕРАТУРА

1. Э. Таненбаум. Современные операционные системы. Пер. с англ. – СПб: Питер, 2010. – 1120 с.
2. LINUX.RU – [Электронный ресурс]: сайт посвященный особенностям работы в среде операционной системы Linux для русскоязычных пользователей – 2012. Режим доступа: <http://www.linux.ru>.
3. Буч, Гради. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ : Пер. с англ. / Гради Буч; Ред. пер. И. Романовский, Ред. пер. Ф. Андреев. - 2-е изд. - М. : БИНОМ, 2000 ; СПб. : Невский Диалект, 2000. - 360 с. : ил.. (1 экземпляр в библиотеке ТУСУРа).
4. Фаулер М. UML. Основы, 3-е издание. – Пер. с англ. – СПб: Символ–Плюс, 2004. – 192 с.,ил.
5. Брайан Керниган, Деннис Ритчи. Язык программирования C: Пер. с англ. / Брайан Керниган, Деннис Ритчи; Пер. В. Бродовой, - М. : Вильямс, 2012. - 304 с.
6. Медианный фильтр — Википедия. [Электронный ресурс]: описание алгоритма медианного фильтра – 2012. Режим доступа: http://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%84%D0%B8%D0%BB%D1%8C%D1%82%D1%80.
7. УМК “Архитектура вычислительных систем” Томск, ТУСУР, УИ 2012 г.