

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

УТВЕРЖДАЮ

Проректор по учебной работе
учреждения образования
«Белорусский государственный
университет информатики и
радиоэлектроники»

_____ В. А. Рыбак
_____ 2023 г.

Регистрационный № УД-_____/уч.

«ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»

**Учебная программа учреждения высшего образования по учебной дисциплине для
специальности**

1-40 02 01 Вычислительные машины, системы и сети

Учебная программа учреждения образования составлена на основе образовательного стандарта ОСВО 1-40 02 01-2021 и учебных планов специальности 1-40 02 01 Вычислительные машины, системы и сети.

СОСТАВИТЕЛИ:

Л. П. Поденок, старший преподаватель кафедры электронных вычислительных машин учреждения образования «Белорусский государственный университет информатики и радиоэлектроники».

РЕЦЕНЗЕНТЫ:

1) Кафедра ПОИТ учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № ____ от _____);

Н. В. Лапицкая, заведующий кафедрой ПОИТ учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент.

2) Объединенный институт проблем информатики НАНН Беларуси.

В. В. Старовойтов, главный научный сотрудник лаборатории идентификации систем, доктор техн. наук, профессор.

РАССМОТРЕНА И РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой электронных вычислительных машин учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № ____ от _____);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № ____ от _____).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа рассчитана на 120 учебных часов (3 з.е.)

План учебной дисциплины в дневной форме получения образования:

Код специальности (направления специальности)	Название специальности (направления специальности)	Курс	Семестр	Аудиторных часов (в соответствии с учебным планом уо)				Академ. часов на курс. работу (проект)	Типовой расчет	Форма промежуточной аттестации
				Всего	Лекции	Лабораторные занятия	Практические занятия, семинары			
1-40 02 01	Вычислительные машины, системы и сети	2	4	68	32	36	-	30	-	экзамен

План учебной дисциплины в заочной форме получения образования:

Код специальности (направления специальности)	Название специальности (направления специальности)	Курс	Семестр	Аудиторных часов (в соответствии с учебным планом уо)				Академ. часов на курс. работу (проект)	Контрольные работы	Форма промежуточной аттестации
				Всего	Лекции	Лабораторные занятия	Практические занятия, семинары			
1-40 02 01	Вычислительные машины, системы и сети	3	6	18	8	8	2	30	-	экзамен

Место учебной дисциплины.

Развитие аппаратных компьютерных средств требует создания системного программного обеспечения, направленного на их эффективное использование. Это обуславливает место курса «Операционные системы и системное программирование» в основе формирования компетенций специалиста в области программного обеспечения отрасли информационных технологий.

Воспитательное значение учебной дисциплины «Операционные системы и системное программирование» заключается в формировании у обучающихся математической культуры и научного мировоззрения; развитии исследовательских умений, аналитических способностей, креативности, необходимых для решения научных и практических задач; развитии познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности; формировании способностей к саморазвитию, самосовершенствованию и самореализации.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

Цель преподавания учебной дисциплины: изучение организации и принципов построения современных операционных систем, основанных на открытых международных стандартах, изучение системных программ, изучение использования аппаратных и программных средств современных процессоров, предназначенных для поддержки многозадачных операционных систем, изучение методологии разработки системно-ориентированных программ (утилит) с использованием современных алгоритмических языков и систем программирования.

Задачи учебной дисциплины:

формирование базовых знаний в области организации и функционирования современных операционных систем;

изучение способов разработки системного программного обеспечения с учетом особенностей современных операционных систем;

овладение методами разработки, тестирования, отладки и документирования программ, направленных на решение системных задач, с использованием современных инструментальных средств.

В результате изучения учебной дисциплины «Операционные системы и системное программирование» формируются следующие компетенции:

универсальные:

владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации

обладать навыками саморазвития и совершенствования в профессиональной деятельности;

проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности;

специализированные:

проектировать системное программное обеспечение на основе знаний об организации функционирования и архитектуры операционных систем ЭВМ.

В результате изучения учебной дисциплины студент (обучающийся) должен:

знать:

- принципы построения операционных систем;
- типовые алгоритмы организации взаимодействия между процессами;
- задачи, решаемые при управлении виртуальной памятью;
- основные системные вызовы стандарта POSIX;
- систему прав доступа в файловых системах ОС UNIX/Linux;
- способы взаимодействия процессов в ОС UNIX/Linux;
- способы межпрограммного взаимодействия с использованием сокетов;
- способы потокового взаимодействия в рамках одного процесса на уровне поддержки стандарта ISO/IEC 14882 2011+ (Язык программирования C++);
- способы потокового взаимодействия POSIX;
- архитектуру и подсистемы ОС UNIX/Linux;
- модель виртуальной памяти процесса;
- понятие совместно используемых объектов в ОС UNIX/Linux;
- средства поддержки многозадачности в ОС UNIX/Linux и методы синхронизации задач;

уметь:

- разрабатывать кроссплатформенные программы в ОС Linux;
- разрабатывать программы, по организации взаимодействия между процессами в рамках стандарта POSIX;
- разрабатывать программы, по организации потокового взаимодействия в рамках стандарта POSIX;
- разрабатывать многопоточные программы с синхронизацией данных в рамках стандарта POSIX;
- разрабатывать совместно используемые объекты (динамически подключаемые библиотеки);
- разрабатывать протоколы и способы межпрограммного взаимодействия;
- использовать проецируемые в память файлы в рамках стандарта POSIX;

владеть:

- современными технологиями проектирования системного программного обеспечения;
- современными технологиями тестирования, отладки, верификации, аттестации и оценки качества системного программного обеспечения;
- методами эффективной эксплуатации системного программного обеспечения.

Перечень учебных дисциплин, усвоение которых необходимо
для изучения данной учебной дисциплины

№ п/п	Название дисциплины	Раздел, темы
1	Программирование на языках высокого уровня	Языки программирования С и С++
2	Программирование на языке ассемблера	Все
3	Основы алгоритмизации и программирования	Все

1. Содержание учебной дисциплины

№ темы	Наименование разделов, тем	Содержание тем
Основы операционных систем		
1	Понятия операционной системы	Операционная система. Классификация ОС. ОС реального времени. Микроядерные и монолитные ОС. Структура ОС. Ядро, командный процессор, подсистема ввода-вывода, система управления памятью, файловая система. Понятие системных вызовов. Системные вызовы стандарта POSIX. Концепция виртуализации.
2	Понятия процесса и потока. Механизмы взаимного исключения	Концепция процесса. Диаграмма состояний процесса. Операции над процессами. Создание и завершение процесса. Иерархия процессов. Структуры управления процессами. Процессы зомби. Реализация процессов в современных ОС. Процессы и потоки. Понятия мультизадачности и многопоточности. Потоки в пространстве пользователя и потоки в ядре. Понятие о прерываниях. Параллельные процессы. Независимые и взаимодействующие процессы. Механизмы уведомления процессов о системных событиях. Взаимодействие процессов. Состояние состязания. Детерминированный набор и условия Бернштейна. Понятие критического ресурса. Критическая секция. Взаимное исключение. Механизмы взаимного исключения. Алгоритмы Деккера, Петерсона, Лэмпорта.
3	Типовые механизмы синхронизации	Операция Test & Set (TS). Поддержка механизма TS в современных процессорах. Семафоры Дейкстры. Базовые операции над семафорами. Мьютексы. Задача «поставщик-потребитель». Инверсия приоритетов. Механизмы синхронизации в современных ОС. Мониторы в языках программирования. Барьеры. Задача «читатели-писатели».
4	Ресурсы. Управление памятью. Организация виртуальной памяти.	Распределение ресурсов, проблема тупиков. Алгоритмы обнаружения тупиков и выхода из них. Требования к управлению памятью. Схемы распределения памяти. Страничная организация памяти. Управление виртуальной памятью – размещение, перемещение, преобразование адресов, замещение. Память процесса. Управление памятью в современных ОС. Защита памяти.
5	Планирование в операционных системах	Стратегии планирования. Дисциплины диспетчеризации. Вытесняющие и невытесняющие алгоритмы. Алгоритмы планирования без переключений. Циклическое и приоритетное планирование. Динамические приоритеты. Планирование в системах реального времени. Планирование потоков. Гарантии обслуживания процесса.
6	Управление вводом-	Организация функций ввода-вывода. Логическая струк-

	выводом и файлами	тура функций ввода-вывода. Буферизация операций ввода-вывода. Дисковое планирование. Система управления файлами. Организация файлов, доступ к файлам. Блочные и символьные операции. Синхронные и асинхронные операции. Отображение ввода-вывода на адресное пространство памяти. Прямой доступ к памяти. Кэширование операций. Упреждающее чтение. Отложенная запись. Программное обеспечение ввода-вывода. Организация драйверов в современных ОС. Псевдоустройства. Файлы устройства.
POSIX-совместимые операционные системы		
7	Введение	Основные принципы организации и построения ОС UNIX. Стандарты SUS и POSIX. Пользователи системы, атрибуты пользователя.
8	Основные утилиты	Системная оболочка. Структура каталогов. Работа с файлами. Управление пользователями и правами. Установка и управление программным обеспечением.
9	Файловая система	Структура файловой системы. Жесткие и символические ссылки. Работа с каталогами, чтение и запись файлов, управление файлами, управление устройствами.
10	Процессы в ОС класса UNIX	Атрибуты процессов. Создание процессов и управление ими. Завершение процессов. Синхронизация процессов. Запуск программы. Средства для управления свойствами процессов. Группы и сеансы.
11	Взаимодействие между процессами I	Сигналы. Обработка сигналов. Неименованные каналы. Именованные каналы. Сообщения, семафоры, совместно используемая память.
12	Потоки POSIX	Атрибуты потоков. Создание и управление потоками. Завершение потоков. Синхронизация потоков. Мьютексы. Переменные состояния. Спинлоки. RW-блокировки.
13	Эффективная работа с файлами	Файлы, отображаемые на память. Неблокирующий и асинхронный ввод-вывод.
14	Системное программирование	Связь ОС UNIX и языка C. Жизненный путь программ. Динамическое связывание. Структура объектного модуля. Системные вызовы и функции стандартных библиотек. Обработка ошибок.
15	Взаимодействие между процессами II	Сокеты. Типы, адреса. Принцип действия и порядок работы. Сокеты AF_LOCAL для локального межпроцессного взаимодействия.
16	Взаимодействие между процессами в сетевой среде	Сокеты AF_INET, AF_INET6, AF_PACKET

2. Информационно-методический раздел

2.1. Литература

2.1.1. Основная

2.1.1.1 Таненбаум, Э. Современные операционные системы / Э. Таненбаум, Х. Бос. – 4-е изд. – Санкт-Петербург : Питер, 2021. – 1120 с. : ил.

2.1.1.2 Стивенс, У. Р. UNIX. Профессиональное программирование / У. Р. Стивенс, С. А. Раго ; пер. А. Киселева. – 2-е изд. – Санкт-Петербург : Символ-плюс, 2007. – 1040 с. : ил.

2.1.1.3 Рочкинд, М. Д. Программирование для UNIX / М. Д. Рочкинд. – 2-е изд., перераб. и доп. – Санкт-Петербург : Русская Редакция : БХВ-Петербург, 2005. – 704 с.

2.1.1.4 Гласс, Г. UNIX для программистов и пользователей / Г. Гласс, К. Эйблс. – 3-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2004. – 848 с. : ил.

2.1.1.5 Реймонд, Э. С. Искусство программирования для Unix / Э. С. Реймонд ; пер. с англ. - Москва : Вильямс, 2005. – 544 с. : ил.

2.1.1.6 Столяров, А. В. Программирование: введение в профессию : в 3 т. Т. II : Системы и сети / А. В. Столяров. – 2-е изд., испр. и доп. – Москва : МАКС Пресс, 2021. – 704 с. : ил.

2.1.1.7 Робачевский, А. М. Операционная система UNIX : учебное пособие [доп. МО РФ] / А. М. Робачевский, С. А. Немнюгин, О. Стестик. – 2-е изд. – Санкт-Петербург : БХВ-Петербург, 2007. – 656 с. : ил.

2.1.1.8 Шоттс, У. Командная строка Linux. Полное руководство / У. Шоттс. – Санкт-Петербург : Питер, 2017. – 480 с.

2.1.1.9 Стивенс, У. Р. UNIX : разработка сетевых приложений / У. Р. Стивенс, Б. Феннер, Э. М. Рудофф. – 3-е изд. – Санкт-Петербург : Питер, 2007. – 1039 с. : ил.

2.1.2. Дополнительная

2.1.2.1 ISO/IEC 9899-2011[2012] – Programming languages – C, 702 p.

2.1.2.2 ISO/IEC 14882-2017[2017] – Programming languages – C++, – 1622 p.

2.1.2.3 IEEE Std 1003.1 -2017 (Revision of IEEE Std 1003.1-2008). IEEE Standard for Information Technology. Portable Operating System Interface (POSIX®). Base Specifications, Issue 7, 2017. – 3951 p.

2.1.2.4 Лав, Р. Ядро Linux. Описание процесса разработки / Р. Лав. – 3-е изд. – Москва : Вильямс, 2013. – 496 с. : ил.

2.1.2.5 ГОСТ 19 Система программной документации (серия стандартов).

2.1.2.6 ГОСТ 2 Система конструкторской документации (серия стандартов).

2.2. Перечень компьютерных программ, наглядных и других пособий, методических указаний и материалов, технических средств обучения, оборудования для выполнения лабораторных работ

Прытков В.А., Уваров А.А., Супонев В.А. Типовые механизмы синхронизации процессов. Учебно-методическое пособие по дисциплине «Системное программное обеспечение ЭВМ» для студентов специальности I-40 02 01 «Вычислительные машины, системы и сети». Мн., БГУИР, 2007.

2.2.1. Технические средства обучения

ПЭВМ с архитектурой x86_64 и объемом ОЗУ 8Гб.

2.2.2. Программное обеспечение

1. Операционная система Linux;
2. Набор утилит binutils;
3. GNU Compiler Collection (C, C++) версия 10 и выше;
4. Ассемблер NASM версия 15.4 и выше;
5. Отладчик GDB;
6. Двухпанельный файловый менеджер;
7. Анализатор использования памяти valgrind.

2.3. Перечень тем практических занятий, их название

Целью практических занятий является закрепление теоретического курса, приобретение навыков решения задач, активизация самостоятельной работы студентов.

№ темы	Наименование практического занятия	Содержание	Обеспеченность по пункту 2.2
8, 14	Основные утилиты	Освоение командной строки оболочки. Создание многомодульной программы и ее анализ с помощью утилит binutils.	2.2.1, 2.2.2

2.4. Перечень тем лабораторных занятий, их название

Основная цель проведения лабораторных занятий состоит в закреплении теоретического материала курса, приобретении навыков программирования с использованием системных вызовов, отладки многопоточных и взаимодействующих программ, оформления отчетов в соответствии с отечественными стандартами.

№ темы по п.1	Наименование лабораторной работы	Содержание	Обеспеченность по пункту 2.2
2, 10	Создание процессов и управление ими	Освоение командной строки оболочки. Базовая работа с процессами средствами оболочки bash.	2.2.3
10, 11	Взаимодействие и синхронизация процессов	Сигналы, семафоры, каналы, совместно используемая память.	2.2.3
2, 12	Создание потоков и управление ими	Базовая работа с потоками. Порождение, взаимодействие, завершение, очистка.	2.2.3
11, 12	Взаимодействие и синхронизация потоков	Сигналы, семафоры, каналы, совместно используемая память.	2.2.3
6, 8, 9	Файловая система	Работа с каталогами. Разработка простого аналога утилиты find	2.2.3
9, 13	Операции с файлами	Работа многопоточного приложения с файлами, отображенными на память,	2.2.3

		барьерная синхронизация.	
15, 16	Взаимодействие процессов. Локальные сокеты.	Кооперация процессов для обработки данных	2.2.3
16	Взаимодействие процессов в сетевой среде.	Реализация системы типа «клиент-сервер» с использованием сокетов	2.2.3

2.5. Курсовая работа, ее характеристика

Курсовой проект по данному предмету — это самостоятельная программная разработка студента по заданной теме в области системного программирования.

Целью курсового проектирования является приобретение навыков в реализации цикла разработки программного средства, начиная от технического задания на проект, анализа предметной области и заканчивая оформлением пояснительной записки и чертежа форматом A1, в соответствии с требованиями действующих стандартов. Особое внимание следует уделить разработке алгоритмов и реализации программного средства.

Возможно выполнение как индивидуальных, так и коллективных курсовых работ (2-3 человека). Курсовая работа выполняется с использованием языков программирования C и C++ (gcc, g++) и языка ассемблера (nasm) для платформы Linux.

В состав курсового проекта входят:

- пояснительная записка;
- графическая часть;
- работающее программное средство.

Пояснительная записка должна отражать основные этапы разработки программного средства.

Перечень тем курсовых работ

- 1) Многопоточная программа для фонового контроля изменений и целостности группы файлов;
- 2) Многопоточная переносимая программа обмена файлами (c, c++);
- 3) Утилита сбора информации о системе (ncurses);
- 4) Простая файловая система (SFS) в пространстве пользователя;
- 5) Утилита форматирования и проверки файловой системы SFS;
- 6) Низкоуровневый редактор блочного устройства уровня секторов (ncurses);
- 7) Анализатор и редактор файловой системы (ncurses);
- 8) Утилита обнаружения и тестирования функций USB-устройства;
- 9) FTP-сервер с возможностью получения/отправки архивированных каталогов;
- 10) FTP-клиент с возможностью получения/отправки архивированных каталогов;
- 11) Разработка и реализация протокола аутентификации клиента на сервере с использованием функционала ssl;
- 12) Разработка и реализация низколатентного протокола сбора неоднородных телеметрических данных от нескольких источников в TCP/IP сети (клиент);
- 13) Разработка и реализация имитационной модели источника неоднородных телеметрических данных (сервер);
- 14) Диспетчер параллельных процессов обработки набора файлов с помощью внешних программ;

- 15) «Корзина» для программ, использующих системный вызов `unlink()`;
- 16) Программа-демонстратор технологии использования стандартных совместно используемых библиотек `.so`, `.dll` из управляемого кода (`mono`, `C#`).
- 17) Утилита контроля появления дубликатов в файловой системе с заменой их на жесткие ссылки и протоколирования фактов замены.
- 18) Разработка субаллокатора памяти в пространстве процесса для обновляемого индекса на основе В-дерева (`c`, `c++`).
- 19) Разработка двухпанельного файлового менеджера (`ncurses`) с вкладками.
- 20) Разработка эмулятора арифметических инструкций с фиксированной запятой.
- 21) Разработка симулятора контроллера без инструкции деления (`nasm`).
- 22) Разработка ассемблера для симулятора контроллера и библиотеки поддержки целочисленных операций.
- 23) Разработка кроссассемблера для симулятора контроллера.
- 24) Разработка симулятора процессора мини-ЭВМ PDP-11 (`nasm`, `c`).
- 25) Разработка ассемблера для симулятора процессора PDP-11 (`c`, `c++`).
- 26) Разработка симулятора DSP slice Virtex 5 (`nasm`, `c`).
- 27) Разработка симулятора DSP slice Virtex 7 (`nasm`, `c`).
- 28) Программа-сторож неправильной раскладки русский/английский с заменой введенного фрагмента и переключением раскладки.
- 29) `ncurses`-оболочка для утилиты `find`.
- 30) Программа-аналог `fdupes` с ограничением поиска и анализа по `mime`-типам файлов.
- 31) Разработка симулятора процессора RAD750.
- 32) Разработка простого ассемблера для процессора RAD750.

3. Перечень рекомендуемых средств диагностики результатов учебной деятельности

Для диагностики результатов учебной деятельности могут использоваться следующие формы:

- 1) реферат по теме в рамках курса;
- 2) отчеты по лабораторным занятиям с их устной защитой.

3.1. Учебно-методическая карта учебной дисциплины в дневной форме получения образования

Номер раздела, темы по п.1	Название раздела, темы	Количество аудиторных часов			Самост оатель- ная работа, часы	Форма контроля знаний
		ЛК	Лаб. зан.	ПЗ		
Основы операционных систем						
1	Понятия операционной системы	2			2	реферат
2	Понятия процесса и потока. Механизмы взаимного исключения	2			2	реферат
3	Типовые механизмы синхронизации.	2			2	реферат
4	Ресурсы, управление памятью. Организация виртуальной памяти.	2			2	реферат
5	Планирование в операционных системах.	2			2	реферат
6	Управление вводом-выводом и файлами.	2			2	реферат
POSIX-совместимые операционные системы						
7	Введение.	2			2	–
8	Основные утилиты.	2	4		4	защита л.р.
9	Файловая система.	2	4		4	защита л.р.
10	Процессы в ОС класса UNIX.	2	4		4	защита л.р.
11	Взаимодействие между процессами I.	2	4		4	защита л.р.
12	Потоки POSIX.	2	4		4	защита л.р.
13	Эффективная работа с файлами.	2	4		4	защита л.р.
14	Системное программирование.	2	4		4	защита л.р.
15	Взаимодействие между процессами II.	2	4		6	защита л.р.
16	Взаимодействие между процессами в сетевой среде.	2	4		4	защита л.р.
	Промежуточная аттестация					экзамен, курсовая работа
	Итого	32	36	-	52	

3.2. Учебно-методическая карта учебной дисциплины в дневной форме получения образования

Номер раздела, темы по п.1	Название раздела, темы	Количество аудиторных часов			Самостоятельная работа, часы	Форма контроля знаний
		ЛК	Лаб. зан.	ПЗ		
Основы операционных систем						
1	Понятия операционной системы	2			2	реферат
2	Понятия процесса и потока. Механизмы взаимного исключения	2			6	реферат
3	Типовые механизмы синхронизации.	2			6	реферат
4	Ресурсы, управление памятью. Организация виртуальной памяти.	2			6	реферат
5	Планирование в операционных системах.				6	реферат
6	Управление вводом-выводом и файлами.				6	реферат
POSIX-совместимые операционные системы						
7	Введение.				2	–
8	Основные утилиты.			2	6	защита л.р.
9	Файловая система.				6	защита л.р.
10	Процессы в ОС класса UNIX.		2		8	защита л.р.
11	Взаимодействие между процессами I.		2		8	защита л.р.
12	Потоки POSIX.		2		8	защита л.р.
13	Эффективная работа с файлами.		2		8	защита л.р.
14	Системное программирование.				8	защита л.р.
15	Взаимодействие между процессами II.				8	защита л.р.
16	Взаимодействие между процессами в сетевой среде.				8	защита л.р.
	Промежуточная аттестация					экзамен, курсовая работа
	Итого	8	8	2	102	

**ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ С ДРУГИМИ
УЧЕБНЫМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ**

Перечень учебных дисциплин	Кафедра, обеспечивающая учебную дисциплину по п. 1	Предложения об изменениях в содержании по изучаемой учебной дисциплине	Подпись заведующего кафедрой, обеспечивающей учебную дисциплину по п.1, с указанием номера протокола и даты заседания кафедры
1	2	3	4
Архитектура персональных компьютеров Основы компьютерных сетей	электронных вычислительных машин	нет	Б. В. Никульшин Протокол № ____ от _____

Заведующий кафедрой электронных
вычислительных машин

_____ Б. В. Никульшин