

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»  
ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ  
КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО УМС ИИКС

Протокол № 8/1/2024

от 28.08.2024 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОС UNIX**

Направление подготовки  
(специальность)

[1] 01.03.02 Прикладная математика и информатика

Семестр	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	В форме практической подготовки/ В	СРС, час.	КСР, час.	Форма(ы) контроля, экз./зач./КР/КП
7	2	72	16	0	32		24	0	3
Итого	2	72	16	0	32	16	24	0	

## АННОТАЦИЯ

В курсе излагаются основные средства, предоставляемые операционной системой UNIX, для создания прикладных и системных многозадачных и многопоточных программ. Особое внимание уделяется специфике создания гибких 64-разрядных приложений. Целью курса является освоение студентами подходов к разработке системных и прикладных программ для операционных систем, реализующих API Single UNIX Specification, включая ОС Linux. Рассматриваются компилятор gcc, средства поддержки раздельной компиляции make, создание библиотек статической и динамической компоновки, интерфейсы доступа к файловой системе. Особое внимание уделяется средствам разработки многопоточных программ, созданию процессов с нужными атрибутами, обработке сигналов. Рассматриваются процессы и сигналы реального времени. Значительное место в курсе уделяется средствам межпроцессного взаимодействия, таким как именованные каналы, разделяемая память, очереди сообщений, а также многочисленным средствам синхронизации – мьютексам, семафорам, условным переменным, блокировкам чтения/записи. Обсуждаются и сравниваются средства SVR4 и POSIX. Изучаются средства RPC – удаленного вызова процедур, а также создания приложений, взаимодействующих по сети. Теоретический материал закрепляется при разработке студентом ряда программ.

### 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины «Разработка программного обеспечения ОС UNIX» являются теоретическое освоение принципов функционирования многозадачных ОС на примере ОС UNIX, проектирования и программирования приложений для UNIX, формирование практических навыков разработки многозадачных и многопоточных прикладных программ, изучения принципов передачи данных между адресными пространствами, синхронизации процессов и потоков, кодирования текстовой информации. Особое внимание уделяется специфике создания гибких 64-разрядных приложений.

### 2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Курс представляет собой дисциплину, которая является частью теоретической профилирующей подготовки студентов. Для успешного освоения дисциплины необходимы знания пользователя операционной системы UNIX, языка программирования Си, архитектуры современных процессоров.

### 3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Универсальные и(или) общепрофессиональные компетенции:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
--------------------------------	--

Профессиональные компетенции в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

Задача профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование профессиональной компетенции; Основание (профессиональный стандарт-ПС, анализ опыта)	Код и наименование индикатора достижения профессиональной компетенции
научно-исследовательский			
анализ и математическое моделирование физических процессов	системы ядерно-энергетического комплекса	ПК-1.1 [1] - способен применять цифровые методы обработки информации  <i>Основание:</i> Профессиональный стандарт: 24.078	З-ПК-1.1[1] - знать методы и алгоритмы компьютерной обработки информации; У-ПК-1.1[1] - уметь обоснованно выбирать алгоритмы при обработке данных; В-ПК-1.1[1] - владеть навыками использования компьютера и/или реализации алгоритмов обработки информации в программном обеспечении
анализ, математическое моделирование динамики систем, разработка законов управления	летательные аппараты	ПК-1.3 [1] - способен анализировать и синтезировать системы автоматического управления  <i>Основание:</i> Профессиональный стандарт: 32.001	З-ПК-1.3[1] - знать методы анализа и синтеза систем автоматического управления; У-ПК-1.3[1] - уметь применять методы теории автоматического управления при разработке киберфизических систем; В-ПК-1.3[1] - владеть навыками использования программного обеспечения для математического моделирования систем автоматического управления
анализ и математическое	системы ядерно-энергетического	ПК-1 [1] - Способен собирать, обрабатывать	З-ПК-1[1] - знать основные методы

<p>моделирование физических процессов</p>	<p>комплекса</p>	<p>и интерпретировать результаты научных исследований в области прикладной математики и информационных технологий</p> <p><i>Основание:</i> Профессиональный стандарт: 24.078</p>	<p>научного познания, методы сбора и анализа информации;; У-ПК-1[1] - уметь анализировать информацию, строить логические схемы, интерпретировать результаты научных исследований, критически мыслить, сравнивать результаты различных исследований, формировать собственную позицию в рамках рассматриваемой задачи;; В-ПК-1[1] - владеть навыками работы с научной литературой и навыками интерпретации результатов научных исследований;</p>
<p>анализ и математическое моделирование физических процессов</p>	<p>системы ядерно-энергетического комплекса</p>	<p>ПК-2 [1] - Способен понимать, применять и совершенствовать современный математический аппарат</p> <p><i>Основание:</i> Профессиональный стандарт: 24.078</p>	<p>3-ПК-2[1] - знать современный математический аппарат, используемый при описании, решении и анализе различных прикладных задач; У-ПК-2[1] - использовать современный математический аппарат для построения математических моделей и алгоритмов решения различных прикладных задач; В-ПК-2[1] - владеть навыками применения современного математического аппарата для построения математических моделей различных</p>

			процессов, для обработки экспериментальных, статистических и теоретических данных, для разработки новых алгоритмов и методов исследования задач различных типов
анализ и математическое моделирование физических процессов	системы ядерно-энергетического комплекса	ПК-3 [1] - Способен осуществлять целенаправленный поиск в сети Интернет и других источниках информации о научных достижениях в области прикладной математики, а также о современных программных средствах, относящихся к предмету исследований  <i>Основание:</i> Профессиональный стандарт: 24.078	З-ПК-3[1] - знать основные референтные базы данных научных публикаций, поисковые системы научной литературы;; У-ПК-3[1] - уметь осуществлять поиск научной литературы с использованием существующих поисковых систем и референтных баз данных;; В-ПК-3[1] - владеть навыками поиска научной литературы;
<b>производственно-технологический</b>			
разработка и сопровождение программного обеспечения	информационные и программные системы	ПК-1.2 [1] - способен разрабатывать и применять прикладные программы при решении задач в области киберфизических и информационных систем  <i>Основание:</i> Профессиональный стандарт: 24.057, Анализ опыта: разработка математического и программного обеспечения киберфизических систем	З-ПК-1.2[1] - знать принципы построения и условия применения программ, используемых в задачах разработки и сопровождения киберфизических и информационных систем; У-ПК-1.2[1] - уметь обоснованно выбирать алгоритмы и программные средства для решения задач проектирования и сопровождения киберфизических и информационных систем; В-ПК-1.2[1] - владеть навыками использования

			прикладных программ при разработке и моделировании киберфизических и информационных систем
разработка математического, программного и алгоритмического обеспечения для анализа и моделирования физических процессов	математические модели процессов в сложных технических системах	ПК-4 [1] - Способен использовать современные языки и методы программирования, комплексы прикладных компьютерных программ, современную вычислительную технику, многопроцессорные вычислительные системы при решении производственных и научно-исследовательских задач в области прикладной математики и информатики  <i>Основание:</i> Профессиональный стандарт: 06.001	3-ПК-4[1] - знать современные языки и технологии программирования, комплексы прикладных компьютерных программ; ; У-ПК-4[1] - уметь разрабатывать наукоемкое программное обеспечение с использованием современных языков программирования ; В-ПК-4[1] - владеть навыками проведения математического моделирования физических процессов с использованием существующих и разработанных программных комплексов

#### 4. ВОСПИТАТЕЛЬНЫЙ ПОТЕНЦИАЛ ДИСЦИПЛИНЫ

Направления/цели воспитания	Задачи воспитания (код)
Профессиональное воспитание	Создание условий, обеспечивающих, формирование ответственности за профессиональный выбор, профессиональное развитие и профессиональные решения (В18)
Профессиональное воспитание	Создание условий, обеспечивающих, формирование научного мировоззрения, культуры поиска нестандартных научно-технических/практических решений, критического отношения к исследованиям лженаучного толка (В19)
Профессиональное воспитание	Создание условий, обеспечивающих, формирование профессионально значимых установок: не производить, не копировать и не использовать программные и технические средства, не приобретённые на законных основаниях; не нарушать признанные нормы

авторского права; не нарушать тайны передачи сообщений, не практиковать вскрытие информационных систем и сетей передачи данных; соблюдать конфиденциальность доверенной информации (В40)

## 5. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Разделы учебной дисциплины, их объем, сроки изучения и формы контроля:

№ п.п	Наименование раздела учебной дисциплины	Недели	Лекции/ Практик. (семинары) / Лабораторные работы, час.	Обязат. текущий контроль (форма*, неделя)	Максимальный балл за раздел**	Аттестация раздела (форма*, неделя)	Индикаторы освоения компетенции
	<i>7 Семестр</i>						
1	Часть 1	1-8	4/4/8		25	КИ-8	3-ПК-2, У-ПК-2, В-ПК-2, 3-ПК-3, У-ПК-3, В-ПК-3, 3-ПК-4, У-ПК-4, В-ПК-4, 3-ПК-1.1, У-ПК-1.1, В-ПК-1.1, 3-ПК-1.2, У-ПК-1.2, В-ПК-1.2, 3-ПК-1.3, У-ПК-1.3, В-ПК-1.3, 3-ПК-1, У-ПК-1, В-ПК-1
2	Часть 2	9-16	4/4/8		25	КИ-16	3-ПК-2, У-ПК-2, В-ПК-2, 3-ПК-3, У-ПК-3, В-ПК-3, 3-ПК-4, У-ПК-4, В-ПК-4,

							3-ПК-1.1, У-ПК-1.1, В-ПК-1.1, 3-ПК-1.2, У-ПК-1.2, В-ПК-1.2, 3-ПК-1.3, У-ПК-1.3, В-ПК-1.3, 3-ПК-1, У-ПК-1, В-ПК-1
	<i>Итого за 7 Семестр</i>		16/0/32		50		
	<b>Контрольные мероприятия за 7 Семестр</b>				50	3	3-ПК-1.1, У-ПК-1.1, В-ПК-1.1, 3-ПК-1.2, У-ПК-1.2, В-ПК-1.2, 3-ПК-1.3, У-ПК-1.3, В-ПК-1.3, 3-ПК-1, У-ПК-1, В-ПК-1, 3-ПК-2, У-ПК-2, В-ПК-2, 3-ПК-3, У-ПК-3, В-ПК-3, 3-ПК-4, У-ПК-4, В-ПК-4

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
КИ	Контроль по итогам
З	Зачет

### КАЛЕНДАРНЫЙ ПЛАН

Недели	Темы занятий / Содержание	Лек., час.	Пр./сем., час.	Лаб., час.
	<i>7 Семестр</i>	16	0	32
<b>1-8</b>	<b>Часть 1</b>	4	4	8

1 - 4	<p><b>ПО поддержки разработки ОС UNIX. Интерфейсы доступа к файловой системе.</b>  ПО поддержки разработки ОС UNIX. Стандарты программирования. Средства поддержки разработки программ и контроля версий. Программы SCCS: принципы работы, основные файлы и команды, соглашение о нумерации версий. Программа–библиотекарь ar. Создание статических библиотек. Компилятор gcc. Основные опции компилятора. Раздельная компиляция. Создание и использование библиотек динамической компоновки. Программа make. Синтаксис make-файла. Макроопределения. Встроенные макроопределения. Фиктивные цели. Суффиксные правила.</p> <p>Стандарты, описывающие интерфейсы программирования в UNIX. Стандарты POSIX и Single UNIX Specification. Особенности и настройка компиляторов C и C++ в UNIX. Компиляторы cc, c89, c99, gcc (GNU). Классификация страниц справочной системы man для разработчика программ.</p> <p>Функции потокового ввода/вывода, разбора аргументов командной строки. Разработка функций с переменным числом аргументов. Функции для работы с регистрационной информацией о пользователях. Реализация системных вызовов и обработка ошибок. Интерфейсы доступа к файловой системе. Системные вызовы для работы с файловой системой. Работа с каталогами и специальными файлами. Блокировки. Права доступа. Жесткие и символические ссылки. Работа с каталогами. Получение информации об атрибутах файла. Блокировка участков файлов. Обязательная и рекомендуемая блокировки. Асинхронный ввод/вывод.</p>	Всего аудиторных часов		
		2	2	4
		Онлайн		
		0	0	0
5 - 8	<p><b>Процессы и сигналы</b>  Процессы и их диспетчеризация. Процессы. Системные вызовы fork, vfork, exec. Родственные и неродственные процессы. Наследование. Ожидание завершения процесса. Классы процессов. Изменение класса и приоритета процессов.</p> <p>Сигналы и их обработка. Сигналы. Способы послыки и обработки сигналов. Надежная обработка сигналов. Задержка и освобождение сигналов. Сигнальная маска. Сигналы реального времени. Передача информации вместе с сигналом. Интервальные таймеры. Таймеры POSIX</p>	Всего аудиторных часов		
		2	2	4
		Онлайн		
		0	0	0
9-16	<b>Часть 2</b>	4	4	8
9 - 12	<p><b>Синхронизация процессов и потоков, межпроцессное взаимодействие.</b>  Межпроцессное взаимодействие. Неименованные каналы. Именованные каналы. Очереди сообщений, семафоры, разделяемая память. Отображение файлов в память. Реализации средств взаимодействия процессов в System V и согласно стандарту POSIX.</p>	Всего аудиторных часов		
		2	2	4
		Онлайн		
		0	0	0

	Легковесные процессы и потоки. Создание потоков и их атрибуты. Синхронизация потоков. Передача управления. Взаимоисключающие блокировки (мьютексы). Условные переменные. Блокировки чтения/записи. Потоки и реентерабельность. Данные потоков.			
13 - 16	<b>Программирование сетевого взаимодействия</b> Двери Sun. Удаленные вызовы процедур. Высокоуровневый интерфейс. Стандартные функции RPC. Язык описания интерфейсов IDL. Встроенные типы данных. Принципы передачи, сериализация, десериализация. Сетевое программирование в UNIX. Гнезда (сокеты). Интерфейс транспортного уровня. Одноранговое взаимодействие. Взаимодействие в системе клиент-сервер. Использование сетевых интерфейсов для межпроцессного взаимодействия.	Всего аудиторных часов		
		2	2	4
		Онлайн		
		0	0	0

Сокращенные наименования онлайн опций:

Обозначение	Полное наименование
ЭК	Электронный курс
ПМ	Полнотекстовый материал
ПЛ	Полнотекстовые лекции
ВМ	Видео-материалы
АМ	Аудио-материалы
Прз	Презентации
Т	Тесты
ЭСМ	Электронные справочные материалы
ИС	Интерактивный сайт

#### ТЕМЫ ЛАБОРАТОРНЫХ РАБОТ

Недели	Темы занятий / Содержание
	<i>7 Семестр</i>
1 - 4	<b>Задание 1</b> Разработать программу, реализующую конвейер, переназначение ввода/вывода и т.п.
5 - 8	<b>Задание 2</b> Разработать программу, использующую сигналы.
9 - 12	<b>Задание 3</b> Разработать программу, использующую средства межпроцессного взаимодействия или могопоточную программу.
13 - 16	<b>Задание 4</b> Разработать программу, использующую средства синхронизации.

#### ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Недели	Темы занятий / Содержание
	<i>7 Семестр</i>
1 - 4	<b>Тема 1</b> Переназначение ввода/вывода. Функции работы с учетной информацией о пользователях.

5 - 8	<b>Тема 2</b> Сигнальные маски. Надежная обработка сигналов. Запуск и ожидание завершения процесса.
9 - 12	<b>Тема 3</b> Многопоточные программы. Мьютексы, семафоры, условные переменные. Разделяемая память.
13 - 16	<b>Тема 4</b> Сокеты.

## 6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Курс базируется на сочетании и совмещении теоретической и практической подготовки студентов в рамках единых занятий. В форме лекции описываются те или иные системные вызовы и особенности архитектуры системы, лежащие в их основе, а затем на семинаре, проводимом в компьютерной лаборатории, решаются задачи, рассмотренные в лекционном курсе. В форме лабораторной работы проводится закрепление пройденного материала посредством разработки программ с использованием описанных средств в компьютерных классах кафедры, оборудованных новейшей вычислительной техникой.

Теоретический материал курса представлен в виде текста лекций.

Темы лабораторных работ разработаны для выработки навыков практического применения системных вызовов при разработке приложений.

## 7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Компетенция	Индикаторы освоения	Аттестационное мероприятие (ЖП 1)
ПК-1	З-ПК-1	З, КИ-8, КИ-16
	У-ПК-1	З, КИ-8, КИ-16
	В-ПК-1	З, КИ-8, КИ-16
ПК-1.1	З-ПК-1.1	З, КИ-8, КИ-16
	У-ПК-1.1	З, КИ-8, КИ-16
	В-ПК-1.1	З, КИ-8, КИ-16
ПК-1.2	З-ПК-1.2	З, КИ-8, КИ-16
	У-ПК-1.2	З, КИ-8, КИ-16
	В-ПК-1.2	З, КИ-8, КИ-16
ПК-1.3	З-ПК-1.3	З, КИ-8, КИ-16
	У-ПК-1.3	З, КИ-8, КИ-16
	В-ПК-1.3	З, КИ-8, КИ-16
ПК-2	З-ПК-2	З, КИ-8, КИ-16
	У-ПК-2	З, КИ-8, КИ-16
	В-ПК-2	З, КИ-8, КИ-16
ПК-3	З-ПК-3	З, КИ-8, КИ-16

	У-ПК-3	3, КИ-8, КИ-16
	В-ПК-3	3, КИ-8, КИ-16
ПК-4	З-ПК-4	3, КИ-8, КИ-16
	У-ПК-4	3, КИ-8, КИ-16
	В-ПК-4	3, КИ-8, КИ-16

### Шкалы оценки образовательных достижений

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-х балльной шкале	Отметка о зачете	Оценка ECTS
90-100	5 – «отлично»	«Зачтено»	A
85-89	4 – «хорошо»		B
75-84			C
70-74			D
65-69	3 – «удовлетворительно»		E
60-64	2 – «неудовлетворительно»	«Не зачтено»	F
Ниже 60			

Оценка «отлично» соответствует глубокому и прочному освоению материала программы обучающимся, который последовательно, четко и логически стройно излагает свои ответы, умеет тесно увязывать теорию с практикой, использует в ответах материалы монографической литературы.

Оценка «хорошо» соответствует твердым знаниям материала обучающимся, который грамотно и, по существу, излагает свои ответы, не допуская существенных неточностей.

Оценка «удовлетворительно» соответствует базовому уровню освоения материала обучающимся, при котором освоен основной материал, но не усвоены его детали, в ответах присутствуют неточности, недостаточно правильные формулировки, нарушения логической последовательности.

Отметка «зачтено» соответствует, как минимум, базовому уровню освоения материала программы, при котором обучающийся владеет необходимыми знаниями, умениями и навыками, умеет применять теоретические положения для решения типовых практических задач.

Оценку «неудовлетворительно» / отметку «не зачтено» получает обучающийся, который не знает значительной части материала программы, допускает в ответах существенные ошибки, не выполнил все обязательные задания, предусмотренные программой. Как правило, такие обучающиеся не могут продолжить обучение без дополнительных занятий.

## **8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **ОСНОВНАЯ ЛИТЕРАТУРА:**

1. ЭИ С 77 Операционные системы : учебник, Староверова Н. А., Санкт-Петербург: Лань, 2022
2. 004 К36 Язык программирования С : , Ритчи Д., Керниган Б., Москва [и др.]: Вильямс, 2013

### **ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:**

1. 004 С80 UNIX: взаимодействие процессов : , Стивенс У.Р., Москва [и др.]: Питер, 2003
2. 004 С80 UNIX: разработка сетевых приложений : , Стивенс У.Р., Москва [и др.]: Питер, 2004
3. 004 Б97 Операционные системы : Параллельные и распределенные системы, Бэкон Д., Харрис Т., Москва [и др.]: ВHV; Питер, 2004
4. 004 И49 Операционные системы : учебное пособие, Илюшечкин В.М., Москва: Бином. Лаборатория знаний, 2009

### **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:**

1. gcc ()
2. ОС Solaris ()
3. putty (компьютерный класс)

### **LMS И ИНТЕРНЕТ-РЕСУРСЫ:**

<https://online.mephi.ru/>

<http://library.mephi.ru/>

## **9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

1. Проектор
2. Компьютерный класс ()

## **10. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ**

Курс посвящен API ОС UNIX и сфокусирован на средствах, закрепленных стандартом. Поэтому рекомендуется ознакомиться со стандартом на сайте [www.unix-system.org](http://www.unix-system.org).

Для освоения программирования с использованием рассматриваемых средств за семестр необходимо написать 4 программы. Средства ОС, которые должны быть использованы в обязательном порядке при написании программы, указаны в заголовке – в теме задания. Для

третьего задания необходимо разработать makefile, но рекомендуется сделать общий makefile для всех заданий.

Учитывая достаточное количество публикаций по программированию в ОС UNIX в сети Интернет, старайтесь выработать навык поиска информации о применении того или иного изучаемого средства.

## **11. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЕЙ**

Студенту предлагается написать 4 программы за семестр. 1 и 2 задания не связаны и могут быть распределены произвольно. 3 и 4 задания следует выдавать одним вариантом, чтобы избежать совпадения тематики. Средства ОС, которые должны быть использованы при написании программы, указаны в заголовке – в теме задания.

Ниже приводятся задания для самостоятельной работы студентов и задания, которые могут быть разобраны на семинарах.

### **Задача 1**

Разработать программу — утилиту, выводящую на экран список всех пользователей системы, список групп, а по запросу – подробную информацию об указанном пользователе, пользователей — членов указанной группы, список групп, в которые входят указанные пользователи.

### **Задача 2**

Разработать программу, дублирующую вывод другой программы в указанный файл.

### **Задача 3**

Разработать программу и соответствующий makefile, который по требованию собирает статическую или динамическую библиотеку и компоует ее с вызывающей программой.

### **Задача 4**

Разработать программу, явно запрашивающую библиотеку динамической компоновки и вызывающую ее функции.

### **Задача 5**

Разработать библиотеку динамической компоновки, подсчитывающую текущее количество ее подключений.

### **Задача 6**

Разработать программу, выдающую собственные характеристики – идентификаторы процесса, процесса-родителя, группы процессов, сеанса, реальный и эффективный пользователя и группы.

### **Задача 7**

Разработать программу, иллюстрирующую написание функции с переменным числом аргументов. Функция выполняет арифметические действия и вызывается так:

Varialc( число, действие, число, ..., действие, число, '=' ).

### **Задача 8**

Разработать программу, выдающую характеристики указываемого файла.

### **Задача 9**

Разработать программу, выдающую содержимое каталога и подсчитывающую число файлов и подкаталогов в нем.

### **Задача 10**

Разработать программу, запускающую указанный процесс на выполнение, а после его завершения сообщаящую статус его завершения.

Задача 11

Разработать программу, запускающую две программы на выполнение, соединяя их в конвейер.

Задача 12

Разработать программу, запускающую программу на выполнение, переназначая ее вывод в задаваемый пользователем файл.

Задача 13

Разработать программы: первая записывает в заданное место указанного файла фразу, введенную пользователем, вторая читает файл с указанного места в заданном объеме. Использовать блокировки.

Задача 14

Разработать программу, организующую передачу вводимых пользователем фраз через неименованный канал между родительским и дочерним процессом.

Задача 15

Разработать программу, создающую неименованный канал между двумя дочерними процессами для передачи пользовательского ввода.

Задача 16

Разработать программу по команде пользователя включающую/отключающую перехват нажатия CTRL-C.

Задача 17

Разработать программу-диспетчер, запускающую две другие, которые выполняются по очереди квантами времени, определяемыми программой-диспетчером.

Автор(ы):

Ктитров Сергей Викторович, к.т.н., доцент