

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО УМС ИИКС

Протокол № 4/1/2023

от 25.04.2023 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, экс./зач./КР/КП
7	2	72	8	8	16		40	0	3
Итого	2	72	8	8	16	16	40	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
ОПК-4 [1] – Способен понимать принципы работы современных информационных технологий и	З-ОПК-4 [1] – Знать основные принципы работы современных информационных технологий и программных средств при решении задач

использовать их для решения задач профессиональной деятельности	<p>профессиональной деятельности</p> <p>У-ОПК-4 [1] – Уметь осуществлять выбор программного средства и применять современные информационные технологии для решения научно-практических задач в профессиональной сфере</p> <p>В-ОПК-4 [1] – Владеть навыками использования современных информационных технологий и программных средств при решении задач профессиональной деятельности</p>
ОПК-5 [1] – Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	<p>З-ОПК-5 [1] – Знать основные языки программирования и методы алгоритмизации, современные технические и программные средства для разработки компьютерных программ</p> <p>У-ОПК-5 [1] – Уметь применять методы алгоритмизации и современные технологии программирования для решения практических задач в различных областях науки и техники</p> <p>В-ОПК-5 [1] – Владеть навыками разработки алгоритмов и компьютерных программ, отладки и тестирования разработанных программных комплексов для решения научно-практических задач</p>

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	<p>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</p> <p><i>Основание:</i> Профессиональный стандарт: 24.078</p>	<p>З-ПК-3[1] - знать основные референтные базы данных научных публикаций, поисковые системы научной литературы;;</p> <p>У-ПК-3[1] - уметь осуществлять поиск научной литературы с использованием существующих поисковых систем и референтных баз данных;;</p> <p>В-ПК-3[1] - владеть навыками поиска научной литературы;</p>

производственно-технологический			
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	З-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)	Воспитательный потенциал дисциплин
Профессиональное воспитание	Создание условий, обеспечивающих, формирование ответственности за профессиональный выбор, профессиональное развитие и профессиональные решения (В18)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования у студентов ответственности за свое профессиональное развитие посредством выбора студентами индивидуальных образовательных траекторий, организации системы общения между всеми участниками образовательного процесса, в том числе с использованием новых информационных технологий.
Профессиональное воспитание	Создание условий, обеспечивающих, формирование научного мировоззрения, культуры поиска нестандартных научно-технических/практических	1. Использование воспитательного потенциала дисциплин/практик «Научно-исследовательская работа», «Проектная практика», «Научный семинар» для: - формирования понимания

	<p>решений, критического отношения к исследованиям лженаучного толка (B19)</p>	<p>основных принципов и способов научного познания мира, развития исследовательских качеств студентов посредством их вовлечения в исследовательские проекты по областям научных исследований. 2.Использование воспитательного потенциала дисциплин "История науки и инженерии", "Критическое мышление и основы научной коммуникации", "Введение в специальность", "Научно-исследовательская работа", "Научный семинар" для:</p> <ul style="list-style-type: none"> - формирования способности отделять настоящие научные исследования от лженаучных посредством проведения со студентами занятий и регулярных бесед; - формирования критического мышления, умения рассматривать различные исследования с экспертной позиции посредством обсуждения со студентами современных исследований, исторических предпосылок появления тех или иных открытий и теорий.
<p>Профессиональное воспитание</p>	<p>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 (B40)</p>	<p>1. Использование воспитательного потенциала дисциплин "Информатика (Основы программирования)", Программирование (Объектно-ориентированное программирование)", "Программирование (Алгоритмы и структуры данных)" для формирования культуры написания и оформления программ, а также привития навыков командной работы за счет использования систем управления проектами и контроля версий. 2.Использование воспитательного потенциала дисциплины "Проектная практика" для формирования культуры решения изобретательских задач, развития логического мышления, путем погружения студентов в научную и</p>

		<p>инновационную деятельность института и вовлечения в проектную работу.</p> <p>3.Использование воспитательного потенциала профильных дисциплин для формирования навыков цифровой гигиены, а также системности и гибкости мышления, посредством изучения методологических и технологических основ обеспечения информационной безопасности и кибербезопасности при выполнении и защите результатов учебных заданий и лабораторных работ по криптографическим методам защиты информации в компьютерных системах и сетях.</p> <p>4.Использование воспитательного потенциала дисциплин "Информатика (Основы программирования)", Программирование (Объектно-ориентированное программирование)", "Программирование (Алгоритмы и структуры данных)" для формирования культуры безопасного программирования посредством тематического акцентирования в содержании дисциплин и учебных заданий.</p> <p>5.Использование воспитательного потенциала дисциплины "Проектная практика" для формирования системного подхода по обеспечению информационной безопасности и кибербезопасности в различных сферах деятельности посредством исследования и перенятия опыта постановки и решения научно-практических задач организациями-партнерами.</p>
--	--	---

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	ЛР-6 (5), ЛР-8 (5), Т-8 (15)	25	КИ-8	З-ОПК-4, У-ОПК-4, В-ОПК-4, 3-ОПК-5, У-ОПК-5, В-ОПК-5, 3-ПК-3, У-ПК-3, В-ПК-3, 3-ПК-4, У-ПК-4, В-ПК-4
2	Часть 2	9-16	4/4/8	ЛР-12 (10), ЛР-16 (5), Т-16 (10)	25	КИ-16	З-ОПК-4, У-ОПК-4, В-ОПК-4, 3-ОПК-5, У-

							ОПК-5, В-ОПК-5, 3-ПК-3, У-ПК-3, В-ПК-3, 3-ПК-4, У-ПК-4, В-ПК-4
	<i>Итого за 7 Семестр</i>		8/8/16		50		
	Контрольные мероприятия за 7 Семестр				50	3	3-ОПК-4, У-ОПК-4, В-ОПК-4, 3-ОПК-5, У-ОПК-5, В-ОПК-5, 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
Т	Тестирование
ЛР	Лабораторная работа
КИ	Контроль по итогам
З	Зачет

КАЛЕНДАРНЫЙ ПЛАН

Недели	Темы занятий / Содержание	Лек., час.	Пр./сем., час.	Лаб., час.
	<i>7 Семестр</i>	8	8	16
1-8	Часть 1	4	4	8
1 - 4	ПО поддержки разработки ОС UNIX. Интерфейсы доступа к файловой системе. 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 ag. 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. 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. 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.	Всего аудиторных часов		
		2	2	4
		Онлайн		
		0	0	0
5 - 8	Процессы и сигналы Процессы и их диспетчеризация. Процессы. Системные вызовы fork, vfork, exec. Родственные и неродственные процессы. Наследование. Ожидание завершения процесса. Классы процессов. Изменение класса и приоритета	Всего аудиторных часов		
		2	2	4
		Онлайн		
		0	0	0

	процессов. Сигналы и их обработка. Сигналы. Способы посылки и обработки сигналов. Надежная обработка сигналов. Задержка и освобождение сигналов. Сигнальная маска. Сигналы реального времени. Передача информации вместе с сигналом. Интервальные таймеры. Таймеры POSIX			
9-16	Часть 2	4	4	8
9 - 12	Синхронизация процессов и потоков, межпроцессное взаимодействие. 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. 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.	Всего аудиторных часов		
		2	2	4
		Онлайн		
		0	0	0
13 - 16	Программирование сетевого взаимодействия 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	Задание 1 Разработать программу, реализующую конвейер, переназначение ввода/вывода и т.п.

5 - 8	Задание 2 Разработать программу, использующую сигналы.
9 - 12	Задание 3 Разработать программу, использующую средства межпроцессного взаимодействия или могопоточную программу.
13 - 16	Задание 4 Разработать программу, использующую средства синхронизации.

ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Недели	Темы занятий / Содержание
	<i>7 Семестр</i>
1 - 4	Тема 1 Переназначение ввода/вывода. Функции работы с учетной информацией о пользователях.
5 - 8	Тема 2 Сигнальные маски. Надежная обработка сигналов. Запуск и ожидание завершения процесса.
9 - 12	Тема 3 Многопоточные программы. Мьютексы, семафоры, условные переменные. Разделяемая память.
13 - 16	Тема 4 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 (КП 1)
-------------	---------------------	-----------------------------------

ОПК-4	З-ОПК-4	З, КИ-8, КИ-16, ЛР-6, ЛР-8, Т-8, ЛР-12, ЛР-16, Т-16
	У-ОПК-4	З, КИ-8, КИ-16, ЛР-6, ЛР-8, Т-8, ЛР-12, ЛР-16
	В-ОПК-4	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-16
ОПК-5	З-ОПК-5	З, КИ-8, КИ-16, ЛР-6, ЛР-8, Т-8, ЛР-12, ЛР-16, Т-16
	У-ОПК-5	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-16
	В-ОПК-5	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-16
ПК-3	З-ПК-3	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-16
	В-ПК-3	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-16
	У-ПК-3	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-16
ПК-4	З-ПК-4	З, КИ-8, КИ-16, ЛР-6, ЛР-8, Т-8, ЛР-12, ЛР-16, Т-16
	У-ПК-4	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-16
	В-ПК-4	З, КИ-8, КИ-16, ЛР-6, ЛР-8, ЛР-12, ЛР-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-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – <i>«отлично»</i>	A	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
85-89	4 – <i>«хорошо»</i>	B	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	
65-69			Оценка «удовлетворительно»

60-64	«удовлетворительно»	Е	выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
Ниже 60	2 – «неудовлетворительно»	Ф	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. ЭИ С 77 Операционные системы : учебник, Санкт-Петербург: Лань, 2019
2. 004 К36 Язык программирования С : , Б. Керниган, Д. Ритчи, Москва [и др.]: Вильямс, 2013

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

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

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.

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 компонуется 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 сообщающую 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

