МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ,
МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

МОО ВО Кыргызско-Российский Славянский университет
имени первого Президента Российской Федерации Б.Н. Ельцина

УТВЕРЖДАЮ

Р.о. декана факультета ЦЕФ
_____ Н. М. Комаров

22.09.2022

Разработка программного обеспечения на базе операционной системы Linux

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	**Информационных и вычислительных технологий**
Учебный план	g090404_25_1пи_рпис.plx Направление подготовки 09.04.04 - РФ, 710400 - КР Программная инженерия Магистерская программа "Разработка программно-информационных систем"
Квалификация	**магистр**
Форма обучения	**очная**
Общая трудоемкость	**4 ЗЕТ**

Часов по учебному плану	144	Виды контроля в семестрах:
в том числе:		зачет с оценкой 4
аудиторные занятия	38	
самостоятельная работа	105,9	

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	**4 (2.2)**		Итого	
Недель	13 2/6			
Вид занятий	УП	РП	УП	РП
Лекции	16	16	16	16
Практические	22	22	22	22
Контактная работа в период теоретического обучения	0,1	0,1	0,1	0,1
В том числе инт.	8	8	8	8
В том числе в форме практ.подготовки	22	22	22	22
Итого ауд.	38	38	38	38
Контактная работа	38,1	38,1	38,1	38,1
Сам. работа	105,9	105,9	105,9	105,9
Итого	144	144	144	144

Программу составил(и):
ст. преп., Гайдамако В.В.;к.т.н., доцент, Верзунов С.Н. _____

Рецензент(ы):
д.т.н., проф. кафедры ИВТ, Лыченко Н.М. _____

Рабочая программа дисциплины
Разработка программного обеспечения на базе операционной системы Linux

разработана в соответствии с ФГОС 3++:
Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.04 Программная инженерия (приказ Минобрнауки России от 19.09.2017 г. № 932)

составлена на основании учебного плана:
Направление подготовки 09.04.04 - РФ, 710400 - КР Программная инженерия
Магистерская программа "Разработка программно-информационных систем"
утвержденного учёным советом вуза от 30.06.2025 протокол № 13

Рабочая программа одобрена на заседании кафедры
Информационных и вычислительных технологий

Протокол от 03.09.2025 г. № 1
Срок действия программы: 2025-2027 уч.г.
Зав. кафедрой д.т.н., проф. Лыченко Н.М.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ
к рабочей программе дисциплины (модуля)

Разработка программного обеспечения на базе операционной системы Linux

наименование дисциплины

№ п/п	Основание внесения изменения	Раздел/ пункт	Содержание изменения
1.	№357-П от 24 июня 2026- О преобразовании Естественно-технического факультета и Факультета архитектуры дизайна и строительства в Политехнический институт КРСУ	Титульный лист. Пункт Разработано	Заменить слова «Разработано Кафедра Информационных и вычислительных технологий» словами «Разработано: Высшая школа Цифровых технологий»
2.			
3.			
4.			
5.			

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Сформировать у обучаемого навыки программирования для ОС Linux на языке C/C++, понимание базовых принципов функционирования операционных систем, управления процессами, управления памятью.
1.2	Освоить базовые утилиты для управления ОС. Освоить работу в командной строке, основы программирования в оболочке bash.
1.3	Ознакомиться с основными IDE для создания приложений в JC Linux – Eclipse, QT.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП	
Цикл (раздел) ООП:	Б1.В.ДВ.02
2.1	**Требования к предварительной подготовке обучающегося:**
2.1.1	Проектирование распределенных информационных систем
2.1.2	Облачные инфраструктуры и сервисы
2.1.3	Проектирование высоконагруженных систем
2.1.4	Тестирование и обеспечение качества программных средств
2.1.5	Теория вычислительных процессов
2.1.6	Принципы WEB - программирования
2.2	**Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:**
2.2.1	Подготовка к защите и защита выпускной квалификационной работы
2.2.2	Производственная практика по получению профессиональных умений и опыта профессиональной деятельности
2.2.3	Преддипломная практика

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ПК-2: Способен проектировать сетевые службы	
Знать:	
Уровень 1	- Принципы структурного и объектно-ориентированного web- программирования; - Базовые научные положения и принципы программной инженерии сопровождения и эволюции ПС; - Методы управления ресурсами Вычислительной Системы (ВС); - Уровни исполнения, интерфейс системных вызовов; - Методы организации взаимодействия процессов (потоков), методы синхронизации; - Методы организации взаимодействия с внешними устройствами.
Уровень 2	Основные виды архитектур и модели обслуживания облачных систем
Уровень 3	- основные методы создания сетевых служб; - протоколы сетей и распределенных систем.
Уметь:	
Уровень 1	- понять поставленную задачу; - выбрать адекватные для её решения метод и технологию или комбинацию методов и технологий; - создавать, отлаживать и тестировать системные приложения и компоненты ядра - получать сведения, анализировать и управлять состоянием системы - организовать взаимодействие процессов, синхронизацию доступа к совместным ресурсам с использованием современных методов
Уровень 2	Применять облачные технологии в конкретных ситуациях
Уровень 3	Проектировать сетевые службы распределенных систем
Владеть:	
Уровень 1	- приёмами проектирования и конструирования web систем для решения базовых задач конкретных предметных областей; - навыками сопровождения программных систем, включая web и распределённые ПС с учётом условий его эксплуатации; - навыками написания системных программ, с использованием системных вызовов; - навыками системного администрирования ОС; - навыками оценки и методами повышения производительности работы системы.
Уровень 2	Навыками разработки программного обеспечения и системного администрирования облачных систем
Уровень 3	Навыками проектирования сетевых служб распределенных систем

В результате освоения дисциплины обучающийся должен

3.1	**Знать:**
3.1.1	- Методы создания программного обеспечения Linux;
3.1.2	- Основные стандарты языка С, стандартные библиотеки языка С, интерфейсы прикладного программного обеспечения (API) POSIX и UNIX;
3.1.3	- Основы управления процессами в параллельной среде – проблемы, методы синхронизации, методы организации взаимодействия процессов;

3.1.4	- Средства обеспечения взаимодействия процессов IPC System V – очереди сообщений, семафоры, разделяемая память, сокеты;
3.1.5	- Принципы взаимодействия с периферийными устройствами, механизм прерываний;
3.1.6	- Основные библиотечные функции и системные вызовы для работы с процессами, памятью, файловой системой;
3.1.7	- Принципы работы, назначение и функции командной оболочки, основы написания скриптов.
3.1.8	
3.2	**Уметь:**
3.2.1	- Написать, скомпилировать и запустить программу на языке С и С++ в командной строке;
3.2.2	- Написать, откомпилировать и выполнить программу в среде Eclipse или QT Community;
3.2.3	- Написать и выполнить скрипт командной оболочки;
3.2.4	- Создавать процессы и отслеживать их состояние, управлять процессами с помощью сигналов, запускать процесс в фоновом и интерактивном режиме, приостанавливать и возобновлять выполнение процесса;
3.2.5	- Организовать работу с файлами и директориями – открытие, закрытие, запись, чтение, получение информации о файле и директории, установку и получение информации о блокировках;
3.2.6	- Создавать клиент-серверные приложения с помощью сокетов;
3.2.7	- Работать с переменными окружения.
3.2.8	
3.3	**Владеть:**
3.3.1	- навыками работы с командной строкой и оконным интерфейсом в .ОС Unix/Linux;
3.3.2	- навыками ручной и автоматизированной сборки программ (с использованием компилятора GCC);
3.3.3	- навыками системного программирования на языке Си в среде ОС Unix/Linux;
3.3.4	- навыками организации взаимодействия процессов с использованием именованных каналов, очередей сообщений, разделяемой памяти;
3.3.5	- навыками работы с файлами и директориями;
3.3.6	- навыками организации взаимодействия процессов на разных машинах с помощью сокетов.
3.3.7	

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетен-ции	Литература	Инте ракт.	Пр. подг.	Примечание
	Раздел 1. Введение в дисциплину. Управление процессами. Работа с файлами.							
1.1	Ознакомление со спецификой сборки ПО в GNU/Linux. Процесс создания исполняемого файла. Динамическая и статическая компоновка Формат ELF. Структура объектного и исполняемого файла. Позиционно-независимый код. Структура разделяемой библиотеки. Типы файлов Индексные дескрипторы и жесткие ссылки Права доступа к файлу. /Лек/	4	4	ПК-2	Л1.1Л2.3 Л2.1 Л2.4 Л2.6 Л2.5Л3.1 Э1 Э2 Э3			
1.2	Командная оболочка и оконный интерфейс. Основные команды работы с файлами, редактор (vi,nano,mcedit)и компилятор, регистрация в системе, написание, компиляция и запуск программы на С и С++ в командной строке. Работа с переменными окружения. /Пр/	4	4	ПК-2	Л1.1Л2.2 Л2.1 Л2.4 Л2.7 Л2.6 Л2.5Л3.1 Э1 Э2 Э3		4	

1.3	Проработка лекционного материала, работа с источниками, установка Linux на виртуальной машине, изучение команд Linux, написание скриптов. /Ср/	4	16	ПК-2	Л1.1Л2.3 Л2.1 Л2.4Л3.1 Э1 Э2 Э3			
1.4	Практическая работа №1. Открытие файла, чтение, запись, блокировка всего файла и областей, получение информации о файле. /Пр/	4	2	ПК-2	Л1.1Л2.2 Л2.1 Л2.4 Л2.6Л3.1 Э1 Э2 Э3		2	
1.5	Проработка лекционного материала, работа с источниками, выполнение практической работы №1. /Ср/	4	16	ПК-2	Л1.1Л2.3 Л2.1 Л2.4 Л2.7Л3.1 Э1 Э2 Э3			
1.6	Обзор механизмов ввода-вывода в Linux (Ubuntu) Файловые дескрипторы Системные вызовы: open, close, write, read и lseek Два способа прочесть содержимое директории Разреженные файлы и специфика их применения Процессы и потоки. Создание и завершение процессов, функции fork,и exec, exit. Управление процессами и потоками. Команды, библиотечные функции и системные вызовы работы с процессами. Получение информации о процессе. айловая система proc. /Лек/	4	4	ПК-2	Л1.1Л2.1 Л2.4Л3.1 Э1 Э2 Э3	1		Интерактивное обсуждение
1.7	Практическая работа №1. Создание процесса, команды ps, top, kill. Функции fork, system, exec, exit. Получение информации о процессе. /Пр/	4	2	ПК-2	Л1.1Л2.2 Л2.1 Л2.5 Л2.4 Л2.6Л3.1 Э1 Э2 Э3		2	
1.8	Практическая работа №1. Инструменты разработчика. Компилятор языка С (gcc). Препроцессор языка С (cpp). Компоновщик объектных модулей (ld). Просмотрщик зависимостей динамической компоновки (ldd). Архиватор библиотек (ar). Система управления сборкой (make). Синтаксис makefile. Просмотровщики содержимого объектных и исполняемых файлов (objdump, nm, readelf). Утилита очистки объектных и исполняемых файлов (strip). Символьный отладчик (gdb). /Пр/	4	2	ПК-2	Л1.1Л2.3 Л2.1 Л2.4 Л2.7Л3.1 Э1 Э2 Э3		2	
1.9	Проработка лекционного материала, работа с источниками, выполнение практической работы №1. /Ср/	4	16	ПК-2	Л1.1Л2.1 Л2.4Л3.1 Э1 Э2 Э3			
	Раздел 2. Взаимодействие процессов.							

2.1	Проблемы при параллельном исполнении. Взаимодействие процессов. Механизмы межпроцессного взаимодействия Linux. Сигналы. Неименованные каналы (pipes). Именованные каналы (named pipes). Сообщения (message queue). Разделяемая память (shared memory). Семафоры (semaphores). /Лек/	4	2	ПК-2	Л1.1Л2.2 Л2.1 Л2.4 Л2.6 Л2.7 Л2.5Л3.1 Э1 Э2 Э3	1		Интерактивное обсуждение
2.2	Практическая работа №2. Сигналы. Неименованные каналы (pipes). Именованные каналы (named pipes). /Пр/	4	2	ПК-2	Л1.1Л2.3 Л2.1 Л2.4Л3.1 Э1 Э2 Э3		2	
2.3	Проработка лекционного материала, работа с источниками, выполнение практической работы №2. /Ср/	4	16	ПК-2	Л1.1Л2.1 Л2.4 Л2.7 Л2.6Л3.1 Э1 Э2 Э3			
2.4	Создание пользовательского интерфейса. Использование псевдографики. GTK. Работа с программными средами Eclipse/QT. Создание переносимых приложений. /Лек/	4	2	ПК-2	Л1.1Л2.2 Л2.1 Л2.4 Л2.5Л3.1 Э1 Э2 Э3	1		Интерактивное обсуждение реализации IPC
2.5	Практическая работа №2. Сообщения (message queue). Разделяемая память (shared memory). Семафоры (semaphores). /Пр/	4	2	ПК-2	Л1.1Л2.3 Л2.1 Л2.4 Л2.6Л3.1 Э1 Э2 Э3		2	
2.6	Проработка лекционного материала, работа с источниками, выполнение практической работы №2. /Ср/	4	16	ПК-2	Л1.1Л2.1 Л2.4 Л2.7Л3.1 Э1 Э2 Э3			
	Раздел 3. Клиент-серверные приложения							
3.1	Взаимодействие процессов на разных машинах. Сокеты. /Лек/	4	2	ПК-2	Л1.1Л2.3 Л2.1 Л2.4 Л2.5Л3.1 Э1 Э2 Э3			
3.2	Практическая работа №3. Создание сервера и клиента с помощью сокетов. /Пр/	4	2	ПК-2	Л1.1Л2.2 Л2.1 Л2.4 Л2.6Л3.1 Э1 Э2 Э3	2	2	Интерактивное обсуждение реализации сокетов
3.3	Проработка лекционного материала, работа с источниками, выполнение практической работы №3. Подготовка к контрольной работе /Ср/	4	11,9	ПК-2	Л1.1Л2.1 Л2.4 Л2.7Л3.1 Э1 Э2 Э3			
	Раздел 4. Создание сценариев командной оболочки.							
4.1	Работа с командной оболочкой – понятие команды и последовательности команд, список команд, перенаправление ввода-вывода, условное исполнение, переменные, аргументы командной строки, написание и запуск скрипта шелла. Sed и awk. Tcl скрипты. Скрипты Perl, Python. /Лек/	4	2	ПК-2	Л1.1 Л1.2Л2.1 Л2.3 Л2.4 Л2.6 Л2.5Л3.1 Э1 Э2 Э3	1		Интерактивное обсуждение

4.2	Практическая работа №4. Создание и запуск скрипта несколькими способами. Переменные, аргументы командной строки. /Пр/	4	2	ПК-2	Л1.1Л2.2 Л2.1 Л2.4Л3.1 Э1 Э2 Э3		2	
4.3	Практическая работа №4. условное исполнение, циклы. Функции. /Пр/	4	2	ПК-2	Л1.1Л2.1 Л2.4 Л2.7Л3.1 Э1 Э2 Э3		2	
4.4	Проработка лекционного материала, работа с источниками, выполнение практической работы №4. /Ср/	4	14	ПК-2	Л1.1Л2.3 Л2.1 Л2.4 Л2.5Л3.1 Э1 Э2 Э3			
4.5	Практическая работа №4. Защита. /Пр/	4	2	ПК-2	Л1.1Л2.2 Л2.1 Л2.4 Л2.7 Л2.6Л3.1 Э1 Э2 Э3	2	2	Интерактивное обсуждение результатов работы.
4.6	Зачет с оценкой /КрТО/	4	0,1	ПК-2	Э1 Э2 Э3			

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

1. Что такое системный вызов?
2. Что такое файл? .Расскажите о файловой системе Linux. Какие типы файлов Linux Вы знаете?
3. Что такое жесткие ссылки? Чем они отличаются от символических?
4. Может ли символ «.» появляться в наименовании файла? Имеет ли он какое-либо особенное значение? Какова максимальная длина имени файлf в Litux?
5. Какие разрешения на использование файла существуют в Litux?
6. Что происходит при открытии файла с помощью системного вызова open()?
7. Системные вызовы для работы с файловой системой. Получение информации о файлах
8. Что такое процесс? Какие структуры существуют в ядре для поддержки процессов?
9. Что такое поток (нить исполнения)? Чем нити отличаются от процессов?
10. Что такое системный вызов?
11. Системные вызовы для работы с сигналами
12. Что такое контекст? Какие виды контекста Вы знаете?
13. Как создаются процессы в Linux? Как завершаются?
14. Расскажите о состояниях процессов
15. Системные вызовы для работы с процессами
16. Что такое процессы-зомби? Как избежать появления зомби?
17. Какие проблемы могут возникнуть при параллельном и псевдопараллельном выполнении процессов и потоков?
18. Для чего нужна синхронизация? Какие способы синхронизации Вам известны?
19. Что такое программный канал (pipe)? Что такое именованные и неименованные каналы, где могут применяться, чем отличаются?
20. API потоков выполнения фирмы Sun и определения POSIX. Синхронизация потоков выполнения: взаимоисключающие блокировки, условные переменные, барьеры, блокировки чтения-записи.
21. Что такое сигналы? Как они обрабатываются?
22. Расскажите о реализации IPC System V для очередей сообщений и разделяемой памяти
23. Опишите работу семафора
24. Каковы достоинства и недостатки многопоточного программирования?
25. Расскажите о приоритетах в Linux. Как изменить приоритет процесса?

Контрольные вопросы для проверки ЗНАТЬ:

1. Что такое системный вызов?
2. Что такое файл? .Расскажите о файловой системе Linux. Какие типы файлов Linux Вы знаете?
3. Что такое жесткие ссылки? Чем они отличаются от символических?
4. Может ли символ «.» появляться в наименовании файла? Имеет ли он какое-либо особенное значение? Какова максимальная длина имени файлf в Litux?
5. Какие разрешения на использование файла существуют в Litux?
6. Что происходит при открытии файла с помощью системного вызова open()?
7. Системные вызовы для работы с файловой системой. Получение информации о файлах
8. Что такое процесс? Какие структуры существуют в ядре для поддержки процессов?
9. Что такое поток (нить исполнения)? Чем нити отличаются от процессов?
10. Что такое системный вызов?
11. Системные вызовы для работы с сигналами
12. Что такое контекст? Какие виды контекста Вы знаете?

13. Как создаются процессы в Linux? Как завершаются?
14. Расскажите о состояниях процессов
15. Системные вызовы для работы с процессами
16. Что такое процессы-зомби? Как избежать появления зомби?
17. Какие проблемы могут возникнуть при параллельном и псевдопараллельном выполнении процессов и потоков?
18. Для чего нужна синхронизация? Какие способы синхронизации Вам известны?
19. Что такое программный канал (pipe)? Что такое именованные и неименованные каналы, где могут применяться, чем отличаются?
20. API потоков выполнения фирмы Sun и определения POSIX. Синхронизация потоков выполнения: взаимоисключающие блокировки, условные переменные, барьеры, блокировки чтения-записи.
21. Что такое сигналы? Как они обрабатываются?
22. Расскажите о реализации IPC System V для очередей сообщений и разделяемой памяти
23. Опишите работу семафора
24. Каковы достоинства и недостатки многопоточного программирования?
25. Расскажите о приоритетах в Linux. Как изменить приоритет процесса?

Контрольные вопросы для проверки УМЕТЬ и ВЛАДЕТЬ:

1. Расскажите о следующих функциях, чем они отличаются?
a. getc() и fgetc()
b. ii. stat() и fstat()
c. iii. printf() и fprintf()
d. iv. scanf() и fscanf()
2. Что такое системный вызов? Расскажите о следующих системных вызовах, чем они отличаются?
a. fork() и vfork()
b. wait() и waitpid()
3. Что делают функции семейства exec?
4. Как проверить состояние процесса? Какие команды для просмотра информации о процессах Вы знаете?
5. Что такое процессы-деймоны, как создать такой процесс?
6. Как посмотреть информацию о процессах?
7. Какие возможности взаимодействия процессов предоставляют программные каналы (pipes) и очереди сообщений? Преимущества и недостатки каждого?
8. Напишите код для создания очереди сообщений
9. Напишите программу, демонстрирующую взаимодействие двух процессов через разделяемую память
10. Напишите, как создать поток с помощью pthread-create?
11. Опишите последовательность функций для создания сокета, использующего TCP протокол
12. Опишите структуру адресов unix и inet сокетов?
13. Что такое скрипт (сценарий) командной оболочки(shell)? Как можно запустить скрипт на выполнение?
14. Что такое «переменная окружения» (environment variable)?
15. Для чего применяется переназначение ввода-вывода?
16. Напишите скрипт, выводящий список файлов, имеющих разрешение на запись и исполнение (в текущей директории)
17. Назначение семейства команд grep, приведите примеры?
18. Что такое awk? Как можно его применять в скриптах? Какие Вы знаете параметры awk?
19. Напишите скрипт sed, выводящий на терминал все строки файла, название которого передается в качестве аргумента командной строки, при этом меняя последовательность символов “abc” на “ABC”
20. Напишите скрипт, распечатывающий аргументы командной строки в обратном порядке.

5.2. Темы курсовых работ (проектов)

Не предусмотрены.

5.3. Фонд оценочных средств

Задания к практическим работам

Практическая работа №1.
Создание консольного приложения из нескольких процессов («читателей» и «писателей») для открытия, записи в файл информации, введенной с клавиатуры и чтения из файла.
Создание и завершение процесса. Функции system и exec. Работа с инструментами – компилятором gcc, компоновщиком объектных модулей ld, просмотрщиком зависимостей динамической компоновки ldd. Архиватор библиотек ar. Система управления сборкой (make). Синтаксис makefile. Проблемы при работе с файлом нескольких процессов. Блокировки файлов.

Практическая работа №2. Взаимодействие процессов.
Использование сигналов для синхронизации. Перехват сигнала. Использование Неименованные каналы (pipes). Именованные каналы (named pipes).
Использование программной среды Eclipse/QT. Настройка программной среды. Отладка.
Очереди сообщения (message queue). Разделяемая память (shared memory). Семафоры (semaphores)
Создать приложение для чтения и записи в разделяемую память, для синзронизации использовать любые методы по выбору.

Практическая работа №3. Клиент-сервер
Создать простой сервер с использованием internet сокетов. Сервис любой, например – показ цитаты, сообщения, интернет страницы, содержимого директории.
Создать клиентское приложение, принимающее адрес и порт сервера в качестве атрибутов командной строки.

Практическая работа №4. Создание скрипта командной оболочки
Переменные, аргументы, awk, sed, условия, циклы, функции

Вопросы к контрольной работе
1. Что такое файл? .Расскажите о файловой системе Linux. Какие типы файлов Linux Вы знаете?
2. Что такое процесс? Какие структуры существуют в ядре для поддержки процессов?
3. Что такое поток (нить исполнения)? Чем нити отличаются от процессов?
4. Что такое системный вызов?
5. Что такое контекст? Какие виды контекста Вы знаете?
6. Как создаются процессы в Linux?
7. Расскажите о состояниях процессов
8. Что такое процессы-зомби? Как избежать появления зомби?
9. Какие проблемы могут возникнуть при параллельном и псевдопараллельном выполнении процессов и потоков?
10. Для чего нужна синхронизация? Какие способы синхронизации Вам известны?
11. Что такое программный канал (pipe)? Что такое именованные и неименованные каналы, где могут применяться, чем отличаются?
12. Что такое сигналы? Как они обрабатываются?
13. Расскажите о реализации IPC UNIX System V для очередей сообщений и разделяемой памяти
14. Опишите работу семафора
15. Каковы достоинства и недостатки многопоточного программирования?

5.4. Перечень видов оценочных средств

Практическая работа; Контрольная работа.
Виды шкал оценивания представлены в Приложении 1.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год
Л1.1	Брайн Керниган, Роб Пайк	UNIX. Программное окружение: Пер. с англ.	Символ-Плюс 2012
Л1.2	Кариев Ч. А.	Разработка Windows-приложений на основе Visual C#: Учебное пособие	Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование 2017

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год
Л2.1	А. Роббинс	Linux: программирование в примерах.: Пер. с англ.	Москва .: Кудиц-Образ 2005
Л2.2	М. Митчелл, Д. Оулдем, А. Самьюэл	Программирование для Linux. Профессиональный подход: Пер. с англ.	Москва .: Издательский дом "Вильямс" 2003
Л2.3	Роббинс А.	Linux: программирование в примерах.: Пер. с англ.	М.: Кудиц-Образ 2005
Л2.4	Павловская Т.А.	C++. Программирование на языке высокого уровня. Для магистров и бакалавров: учебник	СПб.: Питер 2013
Л2.5	Стивенс Р., Раго С.	UNIX. Профессиональное программирование	Символ-Плюс 2014
Л2.6	Роберт Лав	Linux. Системное программирование 2-е издание	СПб. : Питер 2014
Л2.7	Брайан Керниган, Роб Пайк	Практика программирования	МЭ Вильямс 2015

6.1.3. Методические разработки

	Авторы, составители	Заглавие	Издательство, год
Л3.1	Фейгин Я.Д.	Электронный практикум LINUX в примерах: практикум	Бишкек: Изд-во КРСУ 2008

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Введение в прикладное программирование под GNU/Linux	http://iportnov.blogspot.com/2011/10/gnulinux.html
Э2	Unix shell: абсолютно первые шаги	https://habrahabr.ru/post/26

Э3	Ubuntu по-русски	http://ubuntu.ru/

6.3. Перечень информационных и образовательных технологий	
6.3.1 Компетентностно-ориентированные образовательные технологии	
6.3.1.1	Изучение дисциплины студентами осуществляется в форме лекций, лабораторных занятий в аудиторных условиях (лекционные аудитории и компьютерные классы), выполнения заданий на самостоятельную работу, контроля знаний.
6.3.1.2	При проведении лекций используются интерактивные формы обучения.
6.3.1.3	Практические занятия проводятся в компьютерных классах, оснащенных персональными компьютерами с необходимыми параметрами и с установленным необходимым программным обеспечением. Используется Интернет для получения дополнительной информации. Для интерактивной формы обучения используются защита работ перед группой, презентация.
6.3.1.4	Зачет с оценкой проводится в письменной форме в виде ответов на вопросы.
6.3.2 Перечень информационных справочных систем и программного обеспечения	
6.3.2.1	Unix shell: абсолютно первые шаги: https://habrahabr.ru/post/267825/
6.3.2.2	Ubuntu по-русски http://ubuntu.ru/
6.3.2.3	ПРограммное обеспечение
6.3.2.4	Операционная система Linux, компилятор GCC, Eclipse, QT Community

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)	
7.1	Учебная лаборатория компьютерного моделирования и информационных технологийж
7.2	ПК- 13 шт;
7.3	сервер -1; ПК-преподавателя-1;
7.4	Локальная сеть кафедры; Интернет со скоростью 70 Мбит/сек.; зона WI-FI

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)
Технологическая карта дисциплины представлена в Приложении №2. Цель практических занятий – приобретение навыков программирования на языке С/С++ в операционной системе UNIX и создания сценариев командной оболочки. Описание практических работ представлено в электронной папке преподавателя в локальной сети кафедры ИВТ.