

**Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Национальный исследовательский университет «МЭИ»**

Направление подготовки/специальность: 09.04.01 Информатика и вычислительная техника

Наименование образовательной программы: Вычислительные машины, комплексы, системы и сети

Уровень образования: высшее образование - магистратура

Форма обучения: Очная

**Оценочные материалы
по дисциплине
Мультизадачные операционные системы**

**Москва
2023**

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ РАЗРАБОТАЛ:

Разработчик

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Филатов А.В.
	Идентификатор	R48fdeb40-FilatovAV-93eea018

А.В. Филатов

СОГЛАСОВАНО:

Руководитель
образовательной
программы

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Гольцов А.Г.
	Идентификатор	R64210572-GoltsovAG-cebbd3e8

А.Г. Гольцов

Заведующий
выпускающей кафедрой

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Вишняков С.В.
	Идентификатор	R35b26072-VishniakovSV-02810d9

С.В.
Вишняков

ОБЩАЯ ЧАСТЬ

Оценочные материалы по дисциплине предназначены для оценки: достижения обучающимися запланированных результатов обучения по дисциплине, этапа формирования запланированных компетенций и уровня освоения дисциплины.

Оценочные материалы по дисциплине включают оценочные средства для проведения мероприятий текущего контроля успеваемости и промежуточной аттестации.

Формируемые у обучающегося компетенции:

1. ПК-1 Способен осуществлять проектирование вычислительных комплексов и систем, включая разработку аппаратного, программного обеспечения, системную интеграцию, ввод в эксплуатацию

ИД-1 Демонстрирует знание принципов проектирования ЭВМ, микропроцессорных систем и вычислительных систем

ИД-3 Осуществляет разработку аппаратных и программных средств различного назначения в соответствии с техническим заданием

и включает:

для текущего контроля успеваемости:

Форма реализации: Защита задания

1. Защита ЛР №1 (Лабораторная работа)
2. Защита ЛР №2 (Лабораторная работа)
3. Защита ЛР №3 (Лабораторная работа)
4. Защита ЛР №4 (Лабораторная работа)

Форма реализации: Компьютерное задание

1. Операционные системы. ОС UNIX и её командный интерфейс (Тестирование)

Форма реализации: Проверка задания

1. Домашнее задание (Домашнее задание)

БРС дисциплины

1 семестр

Раздел дисциплины	Веса контрольных мероприятий, %						
	Индекс КМ:	КМ-1	КМ-2	КМ-3	КМ-4	КМ-5	КМ-6
	Срок КМ:	4	8	12	14	15	16
Введение. История, особенности и структура ядра UNIX и UNIX - подобных операционных систем. Введение в интерфейсную оболочку Shell							
Введение. История, особенности и структура ядра UNIX и UNIX - подобных операционных систем. Введение в интерфейсную оболочку Shell	+						
Командный интерпретатор Shell							
Командный интерпретатор Shell			+	+			

Подсистема управления файловой системой						
Подсистема управления файловой системой				+		
Подсистема управления процессами, планирование процессов						
Подсистема управления процессами, планирование процессов					+	
Подсистема управления процессами, управление памятью						
Подсистема управления процессами, управление памятью					+	
Подсистема управления процессами, взаимодействие процессов						
Подсистема управления процессами, взаимодействие процессов				+	+	
Подсистема управления вводом-выводом						
Подсистема управления вводом-выводом					+	
Компиляция						
Компиляция						+
Вес КМ:	15	20	20	20	24	1

\$Общая часть/Для промежуточной аттестации\$

СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ТЕКУЩЕГО КОНТРОЛЯ

I. Оценочные средства для оценки запланированных результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Индекс компетенции	Индикатор	Запланированные результаты обучения по дисциплине	Контрольная точка
ПК-1	ИД-1 _{ПК-1} Демонстрирует знание принципов проектирования ЭВМ, микропроцессорных систем и вычислительных систем	Знать: устройство и функционирование файловых систем команды оболочки (интерпретатора) Shell организацию и планирование процессов в ОС устройство и принцип работы компиляторов структурную и функциональную организацию ядра ОС взаимодействие процессов ОС с внешними устройствами и пользователями основы организации взаимодействия процессов с файловой системой стратегии и алгоритмы работы с памятью в ОС Уметь: получать информацию о	Операционные системы. ОС UNIX и её командный интерфейс (Тестирование) Защита ЛР №1 (Лабораторная работа) Защита ЛР №3 (Лабораторная работа) Защита ЛР №4 (Лабораторная работа) Домашнее задание (Домашнее задание)

		текущей конфигурации и настройках системы, настраивать среду ОС работать в командной оболочке интерпретатора Shell	
ПК-1	ИД-3 _{ПК-1} Осуществляет разработку аппаратных и программных средств различного назначения в соответствии с техническим заданием	Знать: алгоритмы порождения процессов и их функционирования программирование сетевого взаимодействия основы разработки сценариев для Shell основы программирования взаимодействия процессов посредством внутренних механизмов ОС Уметь: создавать сценарии (программы-скрипты) в Shell разрабатывать программы с блочным и символьным взаимодействием с файловой системой разрабатывать программные комплексы из процессов, взаимодействующих посредством внутренних механизмов ОС	Защита ЛР №2 (Лабораторная работа) Защита ЛР №3 (Лабораторная работа) Защита ЛР №4 (Лабораторная работа)

II. Содержание оценочных средств. Шкала и критерии оценивания

КМ-1. Операционные системы. ОС UNIX и её командный интерфейс

Формы реализации: Компьютерное задание

Тип контрольного мероприятия: Тестирование

Вес контрольного мероприятия в БРС: 15

Процедура проведения контрольного мероприятия: Тестирование в системах Webex или Прометей

Краткое содержание задания:

Ответы на вопросы теста

Контрольные вопросы/задания:

Знать: структурную и функциональную организацию ядра ОС	1. Укажите, какое ПО является системным: *компиляторы - офисы *драйверы *операционные системы 2. Укажите функции ОС: - управление БД *управление памятью - управление процессами *компиляция *администрирование 3. Укажите, какие функции имеются в Shell: *программный язык - планировщик процессов *командный интерфейс *генерация имен файлов - структурирование файлов 4. Укажите верно написанные (в командной строке) переназначения ввода-вывода: *> 2< *3>> *> 5 *1> 5. Какие права доступа регулярному файлу file1 даёт команда <code>chmod 534 file1</code>? -rw--wxr-x - drw-rwxr-- -rw-r-x-wx - dr-xr-xr-x *-r-x-wxr-- 6. Какие права доступа каталогу dir2 дала команда <code>chmod g-w o-x dir2</code>? -rwr-xr-x - drwxrwxr-- -r--wxr-x
--	--

	<pre>*drwxr-x--- drwx-w--w- 7.Какое содержимое будет в переменной BB в результате выполнения?: AA=Privet BB='echo \$AA' Privet echo Privet *echo \$AA echo AA echo \$Prive 8.Для файлов с какими атрибутами следующая команда даст положительный (0-й) код возврата?: test -f file drwxrwxrwx *-r--r--r-- lwxrwxrwx brw-rw-rw- *-rwx-----</pre>
--	---

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 80

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 40

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

Оценка: 2

Описание характеристики выполнения знания: Оценка "неудовлетворительно" выставляется если задание выполнено неверно или преимущественно не выполнено

КМ-2. Защита ЛР №1

Формы реализации: Защита задания

Тип контрольного мероприятия: Лабораторная работа

Вес контрольного мероприятия в БРС: 20

Процедура проведения контрольного мероприятия: Выполнение задания на ЛР на персональном компьютере с установленной UNIX или UNIX-подобной (например, Linux) ОС или на удалённой системе, и защита её

Краткое содержание задания:

Выполнить лабораторную работу и защитить её

Контрольные вопросы/задания:

Знать:	команды	оболочки	1.Могут ли одновременно в ОС типа UNIX
--------	---------	----------	--

(интерпретатора) Shell	существовать два пользователя с одним и тем же именем, но работающие на разных терминалах 2. Могут ли одновременно в ОС типа UNIX существовать два обычных разноименных пользователя, запускающие свои задания с одного и того же терминала 3. Может ли один и тот же пользователь одновременно выполнять более одного процесса и при этом каждый процесс в своей директории? 4. Какие команды из изученных в данной лабораторной работе возможно объединить через конвейер 5. Какие команды из изученных в данной лабораторной работе допускают переназначение ввода и каким образом? 6. К чему могло бы привести выполнение команды <code>chmod u-rwx</code> ?
Уметь: получать информацию о текущей конфигурации и настройках системы, настраивать среду ОС	1. Продемонстрируйте просмотр информации о зарегистрированных пользователях и поясните её.
Уметь: работать в командной оболочке интерпретатора Shell	1. Продемонстрируйте выполнение команд создания, переноса, копирования, просмотра содержимого, изменение прав доступа и удаления файла и каталога. 2. Продемонстрируйте переназначение ввода-вывода

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

Оценка: 2

Описание характеристики выполнения знания: Оценка "неудовлетворительно" выставляется если задание выполнено неверно или преимущественно не выполнено

КМ-3. Защита ЛР №2

Формы реализации: Защита задания

Тип контрольного мероприятия: Лабораторная работа

Вес контрольного мероприятия в БРС: 20

Процедура проведения контрольного мероприятия: Выполнение задания на ЛР на персональном компьютере с установленной UNIX или UNIX-подобной (например, Linux) ОС или на удалённой системе, и защита её

Краткое содержание задания:

Выполнить лабораторную работу и защитить её

Контрольные вопросы/задания:

Знать: основы разработки сценариев для Shell	1. Как определен в Bourne-shell файл начала сеанса? 2. Как называются командные файлы в Bourne-shell? 3. Каким образом устанавливаются каталоги обязательного поиска команд в Bourne-shell? 4. Какие возможности группирования команд Вам известны в Bourne-shell, каким образом они определены для интерпретатора? 5. Каким образом выполнить Shell-файл в Bourne-shell? 6. Каким образом может быть произведено построение операторов цикла в Bourne-shell? 7. Каким образом можно построить многоальтернативный выбор в Bourne-shell?
Уметь: создавать сценарии (программы-скрипты) в Shell	1. Продемонстрируйте выполнение созданного вами (согласно варианту задания) сценария (скрипта), поясните его программный текст и результаты выполнения 2. Продемонстрируйте завершения одного скрипта другим

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто, выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

Оценка: 2

Описание характеристики выполнения знания: Оценка "неудовлетворительно" выставляется если задание выполнено неверно или преимущественно не выполнено

КМ-4. Защита ЛР №3

Формы реализации: Защита задания

Тип контрольного мероприятия: Лабораторная работа

Вес контрольного мероприятия в БРС: 20

Процедура проведения контрольного мероприятия: Выполнение задания на ЛР на персональном компьютере с установленной UNIX или UNIX-подобной (например, Linux) ОС или на удалённой системе, и защита её

Краткое содержание задания:

Выполнить лабораторную работу и защитить её

Контрольные вопросы/задания:

Знать: основы организации взаимодействия процессов с файловой системой	1. Какие действия выполняет UNIX при открытии и закрытии файла
Знать: устройство и функционирование файловых систем	1. В чем особенности работы с библиотечными функциями для буферизованных обменов с файлами? 2. Сравните системные вызовы для блочного и символьного обмена данными. Когда выгодно использовать каждый из них?
Знать: программирование сетевого взаимодействия	1. В чем особенности системных вызовов для побайтового обмена с файлами?
Уметь: разрабатывать программы с блочным и символьным взаимодействием с файловой системой	1. Написать две программы, одну, использующую библиотеку F.H, другую, использующую библиотеку L.H согласно варианту задания и продемонстрируйте их выполнение 2. Используя GDB, проведите отладку программ. Пропредмонстрируйте работу команд break, watch, print

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

Оценка: 2

Описание характеристики выполнения знания: Оценка "неудовлетворительно" выставляется если задание выполнено неверно или преимущественно не выполнено

КМ-5. Защита ЛР №4

Формы реализации: Защита задания

Тип контрольного мероприятия: Лабораторная работа

Вес контрольного мероприятия в БРС: 24

Процедура проведения контрольного мероприятия: Выполнение задания на ЛР на персональном компьютере с установленной UNIX или UNIX-подобной (например, Linux) ОС или на удалённой системе, и защита её

Краткое содержание задания:

Выполнить лабораторную работу и защитить её

Контрольные вопросы/задания:

Знать: взаимодействие процессов ОС с внешними устройствами и пользователями	1.Как в программе переназначить ввод/вывод из/в устройства в файл или канал или наоборот?
Знать: организацию и планирование процессов в ОС	1.В каких состояниях может находиться процесс?
Знать: стратегии и алгоритмы работы с памятью в ОС	1.Что такое swapping и paging? Как организуется paging?
Знать: алгоритмы порождения процессов и их функционирования	1.Что такое процесс и чем он отличается от программы? 2.Как порождаются процессы в ОС UNIX?
Знать: основы программирования взаимодействия процессов посредством внутренних механизмов ОС	1.Опишите формат системного вызова PIPE и особенности его работы 2.Что такое сигналы и как они используются в ОС UNIX? 3.В чем особенности и отличия поименованных и непоименованных каналов?
Уметь: разрабатывать программные комплексы из процессов, взаимодействующих посредством внутренних механизмов ОС	1.Опишите работу системных вызовов KILL и SIGNAL, продемонстрируйте их использование в программе 2.Продемонстрируйте в программе работу с непоименованным каналом, его создание и использование 3.Продемонстрируйте в программе создание (порождение) нового процесса и поясните эту процедуру 4.Организуйте взаимодействие двух процессов через непоименованный канал путём переназначения стандартного вывода одного процесса с экрана на канал и переназначения стандартного ввода другого процесса с клавиатуры на канал, т.е. организовав конвейер в Shell.

Описание шкалы оценивания:

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Оценка "отлично" выставляется если задание выполнено в полном объеме или выполнено преимущественно верно

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Оценка "хорошо" выставляется если большинство вопросов раскрыто. выбрано верное направление для решения задач

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Оценка "удовлетворительно" выставляется если задание преимущественно выполнено

Оценка: 2

Описание характеристики выполнения знания: Оценка "неудовлетворительно" выставляется если задание выполнено неверно или преимущественно не выполнено

КМ-6. Домашнее задание

Формы реализации: Проверка задания

Тип контрольного мероприятия: Домашнее задание

Вес контрольного мероприятия в БРС: 1

Процедура проведения контрольного мероприятия: Опрос на лекции. 10 минут.

Краткое содержание задания:

Расскажите про синтаксический и семантический анализ программы.

Контрольные вопросы/задания:

Знать: устройство и принцип работы компиляторов	1.Что такое БНФ-формы?
---	------------------------

Описание шкалы оценивания:

Оценка: зачтено

Описание характеристики выполнения знания: Оценка "зачтено" выставляется если задание выполнено правильно или с незначительными недочетами

Оценка: не зачтено

Описание характеристики выполнения знания: Оценка "не зачтено" выставляется если задание не выполнено в отведенный срок или результат не соответствует заданию

СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1 семестр

Форма промежуточной аттестации: Экзамен

Пример билета

1. Коллективные передачи данных в *MPI*. Виды коллективных операций. Барьерная синхронизация, распределение и сбор данных. Примеры.
2. Классификации вычислительных систем (Флина, Хокни, по доступу к памяти). Реконфигурируемые вычислительные системы (PBC) на ПЛИС. Концепция, архитектура базового модуля PBC, структуры компонентов и системы в целом. Требования к элементной базе.
3. Задача. Реализовать выполнение фрагмента программы (гнезда циклов) пятью потоками *OpenMP*. Использовать директиву *section*.

```
for (i=17; i<42; i++)  
for (j=10; j<50; j++)  
{ a[i][j]=a[i][j]+a[(int) i div 8][(int) j div 8]*x+18;  
  a[i][j]+=a[i-1][j]*b[i]; s+=a[i][j];  
}
```

Процедура проведения

На подготовку 60 минут. Ответ в устном виде.

I. Перечень компетенций/индикаторов и контрольных вопросов проверки результатов освоения дисциплины

1. Компетенция/Индикатор: ИД-1ПК-1 Демонстрирует знание принципов проектирования ЭВМ, микропроцессорных систем и вычислительных систем

Вопросы, задания

- 1.Операционные системы. Назначение операционных систем. Основные сведения об ОС *UNIX*. Структура ядра ОС *UNIX*. Командный интерпретатор *Shell*
- 2.Командный интерпретатор *Shell*. Основные задачи *Shell*. Семейство интерпретаторов *Shell*. Выполнение команд оболочкой *Shell*. Переменные окружения. Атрибуты пользователя. Метасимволы
- 3.Командный интерпретатор *Shell*. Выполнение команд оболочкой *Shell*. Формат команды. Способы объединения команд в строке. Метасимволы. Преобразование командной строки. Генерация имен файлов
- 4.Командный интерпретатор *Shell*. Основные задачи *Shell*. Порты ввода/вывода команд и скриптов. Переназначение портов ввода/вывода. Программирование скриптов на *Shell*. Компоненты скрипта. Параметры командной строки
- 5.Командный интерпретатор *Shell*. Основные задачи *Shell*. Формат команды. Команды по работе с каталогами. Порты ввода/вывода команд и скриптов. Переназначение портов ввода-вывода
- 6.Командный интерпретатор *Shell*. Выполнение команд оболочкой *Shell*. Формат команды. Типы файлов в *UNIX*-системах. Команды по работе с файлами. Атрибуты файлов. Права доступа

- 7.Подсистема управления файловой системой. Типы файловых систем. Логическая структура файловой системы ОС *UNIX*. Виртуальная файловая система *VFS*. Монтирование ФС и каталогов. Функции для работы с ФС
- 8.Подсистема управления файловой системой. Типы файловых систем. Структура файловой системы *s5fs*. Суперблок *s5fs*. Структура *inode* в *s5fs*. Адресация данных. Структура каталога *s5fs*. Жёсткие и символьные ссылки. Схема доступа к данным файлов в *s5fs*
- 9.Подсистема управления файловой системой. Монтирование ФС и каталогов. Файловая система *FFS*. Структура файловой системы *FFS*. Структура каталога *FFS*. Взаимодействие процессов с файлами разных ФС
- 10.Виртуальная файловая система *VFS*. Взаимодействие процессов с файлами разных ФС. Коммутация файловых систем. Монтирование ФС и каталогов
- 11.Виртуальная файловая система *VFS*. Взаимодействие процессов с файлами разных ФС. Доступ к содержимому файлов с помощью *VFS*
- 12.Файловые системы *Ext* (*Extfs*). Структура файловой системы *Ext2*. Структура суперблока *Ext2*. Некоторые поля структуры *inode Ext2*. Структура элемента каталога *Ext2*. Определение группы блоков по номеру *inode* и его позиции в группе. Журналирование в *Ext3*. Отличия *Ext4* от *Ext3*
- 13.Управление процессами. Состояния процесса. Алгоритмы планирования назначения процессов на процессор. Расчёт приоритета процесса (по алгоритму *FSS*). Команда просмотра информации о процессах *ps*
- 14.Подсистема управления памятью. Функции подсистемы управления памятью. Стратегии подсистемы управления памятью. Стратегии загрузки разделов. Стратегии размещения разделов, достоинства и недостатки. Методы трансляции адресов
- 15.Подсистема управления памятью. Стратегии подсистемы управления памятью. Стратегии загрузки разделов. Методы трансляции адресов. Схема трансляции виртуального адреса в физический с использованием сегментного и страничного механизмов
- 16.Подсистема управления памятью. Методы трансляции адресов. Стратегии замены страниц. Конфликты из-за страниц. Алгоритм замены страниц согласно принципу Денинга
- 17.Подсистема управления вводом-выводом. Взаимодействие процессов с периферийными устройствами. Драйверы. Функции драйвера. Системный буферный кэш и символьная очередь
- 18.Подсистема управления вводом-выводом. Взаимодействие процессов с периферийными устройствами. Управление блочными устройствами. Доступ к блочному устройству и его данным
- 19.Подсистема управления вводом-выводом. Взаимодействие процессов с периферийными устройствами. Терминальные линии
- 20.Способы трансляции программ. Компиляция. Структура компилятора. Этапы компиляции. Лексический и синтаксический анализ. Генерация кода
- 21.Компиляция. Этапы компиляции. Лексический, синтаксический и семантический анализ. Оптимизация. Генерация кода

Материалы для проверки остаточных знаний

- 1.Что делает команда Shell *mkdir*?

Ответы:

Сформулировать ответ

Верный ответ: Создает каталог

- 2.Что делает команда Shell *ls* с модификаторами *-i -l -a*?

Ответы:

Сформулировать ответ

Верный ответ: Выводит содержимое каталога, показывая скрытые файлы (модификатор -a), в длинном формате (модификатор -l) и номером индексного дескриптора (inode) каждого файла (модификатор -i)

3. Что такое inode?

Ответы:

Сформулировать ответ

Верный ответ: Inode (индексный дескриптор) - описатель файла в индексируемых файловых системах, содержащий метаданные файла и ссылки на блоки для хранения данных (содержимого файла)

4. Что такое VFS?

Ответы:

Сформулировать ответ

Верный ответ: VFS (Virtual file switch/system) - виртуальный файловый коммутатор или виртуальная файловая система. Это промежуточный слой в подсистеме управления файловой системой между процессами и конкретными конечными физическими файловыми системами разных типов, подключёнными (смонтированными) в данный момент к системе

5. Для чего нужен специальный файл устройства?

Ответы:

Сформулировать ответ

Верный ответ: В UNIX-системах за внешними устройствами закреплены специальные файлы-устройства. Используя эти файлы, процессы могут работать (открывать/закрывать, читать/писать, диагностировать, управлять) с устройствами как с файлами, точнее посредством этих файлов

2. Компетенция/Индикатор: ИД-3ПК-1 Осуществляет разработку аппаратных и программных средств различного назначения в соответствии с техническим заданием

Вопросы, задания

1. Программирование скриптов на *Shell*. Компоненты скрипта. Параметры командной строки. Переменные окружения в скрипте. Порты ввода/вывода. Переназначение портов ввода/вывода. Управляющие программные структуры в скриптах
2. Подсистема управления процессами. Процессы в *UNIX*-системах. Типы процессов в *UNIX*-системах. Образ и изображение процесса. Основные атрибуты процесса. Состояния процесса. Порождение процессов
3. Подсистема управления процессами. Типы процессов в *UNIX*-системах. Образ и изображение процесса. Дескриптор процесса в таблице процессов (*proc*). Структура *u_area*. Контекст процесса
4. Управление процессами. Жизненный цикл процесса. Порождение процессов, системный вызов *fork*. Что наследует и не наследует потомок от предка. Контекст процесса. Замена контекста процесса. Функции семейства *exec*
5. Механизмы взаимодействия процессов. Очереди сообщений. Функции работы с очередями сообщений. Потоки *threads*
6. Механизмы взаимодействия процессов. Разделяемая память. Функции работы с разделяемой памятью
7. Механизмы взаимодействия процессов. Семафоры. Операции над семафорами. Функции работы с наборами семафоров
8. Механизмы взаимодействия процессов. Непоименованные и поименованные каналы, создание и работа с каналами
9. Механизмы взаимодействия процессов. Сигналы. Функции и работа с сигналами
10. Механизмы взаимодействия процессов. Сокеты. Функции сокеты. Типы сокеты Беркли. Семейства сокеты (домены). Протоколы. Поточные сокеты и работа с ними

11. Механизмы взаимодействия процессов. Сокеты. Функции сокеты. Типы сокеты Беркли. Семейства сокеты (домены). Протоколы. Датаграммы сокеты и работа с ними

Материалы для проверки остаточных знаний

1. Как порождается новый процесс?

Ответы:

Сформулировать ответ

Верный ответ: Новый процесс - потомок, порождается от уже имеющегося процесса - предка, путём выполнения последним системного вызова FORK (вызова функции fork()). После вызова FORK процессом предком, завершают его уже два процесса - предок и потомок, которые продолжают выполнять одну и ту же программу (что выполнял процесс предок) с момента вызова FORK.

2. Как процесс узнаёт что он вновь порождённый процесс-потомок?

Ответы:

Сформулировать ответ

Верный ответ: По окончании системного вызова FORK, функция fork() возвращает вызвавшему её процессу-предку идентификатор (значение больше нуля) процесса потомка, а процессу-потомку ноль.

3. Для чего нужны функции семейства get?

Ответы:

Сформулировать ответ

Верный ответ: Функции семейства get позволяют процессу получить доступ к экземпляру механизма взаимодействия процессов. msgget - к очереди сообщений, shmget - к сегменту разделяемой памяти, semget - к набору семафоров

II. Описание шкалы оценивания

Оценка: 5

Нижний порог выполнения задания в процентах: 70

Описание характеристики выполнения знания: Работа выполнена в рамках "продвинутого" уровня. Ответы даны верно, четко сформулированные особенности практических решений

Оценка: 4

Нижний порог выполнения задания в процентах: 60

Описание характеристики выполнения знания: Работа выполнена в рамках "базового" уровня. Большинство ответов даны верно. В части материала есть незначительные недостатки

Оценка: 3

Нижний порог выполнения задания в процентах: 50

Описание характеристики выполнения знания: Работа выполнена в рамках "порогового" уровня. Основная часть задания выполнена верно. на вопросы углубленного уровня

Оценка: 2

Описание характеристики выполнения знания: Работа не выполнена или выполнена преимущественно неправильно

III. Правила выставления итоговой оценки по курсу