

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Петровская Анна Викторовна

Должность: Директор

Дата подписания: 28.02.2025 09:52:35

Уникальный программный ключ:

798bda6555fbdebe827768f6f1710bd17a9070c31dc106abac5aff10c8e519f

Приложение 6

к основной профессиональной образовательной программе
по направлению подготовки 09.03.03 «Прикладная информатика»
направленность (профиль) программы «Прикладная информатика
в экономике»

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский экономический университет имени Г.В. Плеханова»

Краснодарский филиал РЭУ им. Г. В. Плеханова

Факультет экономики, менеджмента и торговли

Кафедра экономики и цифровых технологий

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине **Операционные системы**

Направление подготовки

09.03.03 Прикладная информатика

Направленность (профиль) программы

Прикладная информатика в экономике

Уровень высшего образования

Бакалавриат

Год начала подготовки 2024

Краснодар – 2024 г.

Составитель:

к.к., доцент кафедры экономики и цифровых технологий Л.В. Кухаренко

Оценочные материалы одобрены на заседании кафедры экономики и цифровых технологий Краснодарского филиала РЭУ им. Г.В. Плеханова протокол № 9 от 14 марта 2024 г.

Оценочные материалы составлены на основе рабочей программы по дисциплине «Операционные системы», утвержденной на заседании базовой кафедры цифровой экономики и развития информационного общества федерального государственного бюджетного образовательного учреждения высшего образования «Российский экономический университета имени Г.В. Плеханова» протокол № 11 от 17 мая 2021 г., разработанной авторами:

Коньковым М.Н., к.э.н., доцентом, базовой кафедры цифровой экономики и развития информационного общества

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по учебной дисциплине «Операционные системы»

ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ И ЭТАПОВ ИХ ФОРМИРОВАНИЯ ПО ДИСЦИПЛИНЕ

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций (код и наименование индикатора)	Результаты обучения (знания, умения)	Наименование контролируемых разделов и тем
ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2. Осуществляет установку и конфигурирование операционных систем	<p>ОПК-5.2. 3-1. Знает современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды</p> <p>ОПК-5.2. 3-2. Знает объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом</p> <p>ОПК-5.2. 3-3. Знает принципы построения операционных систем</p>	<p>Тема 1 Введение. Основные определения и понятия. Назначение, функции и классификация операционных систем.</p> <p>Тема 2 Архитектура операционных систем.</p> <p>Тема 3 Управление мультизадачным вычислительным процессом.</p> <p>Тема 4 Управление памятью. Методы, алгоритмы и средства.</p>
		<p>ОПК-5.2. У-1. Умеет осуществлять установку и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом</p> <p>ОПК-5.2. У-2. Умеет создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий</p>	<p>Тема 5 Файловые системы.</p> <p>Тема 6 Подсистема ввода-вывода.</p> <p>Тема 7 Информационная безопасность операционных систем.</p>

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ХАРАКТЕРИЗУЮЩИЕ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

Перечень учебных заданий на аудиторных занятиях

Индикаторы достижения: ОПК-5.2

Задания, выполняемые на аудиторных занятиях по темам 1-7

Задание 1.1

- 1) Вывести на экран все свойства объекта (Property) который, описывает процессы, выполняемые на компьютере (Get-Process).
- 2) Вывести на экран все методы (Method) работы с процессами.
- 3) Вывести на экран свойства процессов и методы работы с ними.

Задание 1.2

- 1) Найти директорию, где имеется достаточно файлов.
- 2) Вывести содержимое найденной директории. Список выводимых свойств: CreationTime, Name, Mode, Length.

Задание 1.3

- 1) Найти директорию, где имеется достаточно файлов.
- 2) Вывести содержимое найденной директории. Список выводимых свойств: CreationTime, Name, Mode, Length.
- 3) Вывести на экран информацию из п.2 упорядочив по времени создания объекта (CreationTime).
- 4) Вывести на экран информацию из п.2 упорядочив её по имени (Name) в обратном порядке (параметр Descending).

Задание 1.4

- 1) Вывести список, установленного на компьютере ПО. Список выводимых свойств: Name, Vendor, InstallDate.
- 2) Упорядочить выводимую информацию по дате инсталляции ПО (InstallDate) по убыванию (параметр Descending).
- 3) Для п.2, только 5 последних установок.
- 4) Для п.3, добавить 3 самые ранние установки.

Задание 1.5

- 1) Вывести список процессов (Get-Process).
- 2) Посмотреть свойства, описывающие процессы (Get-Member)
- 3) Вывести только имя процесса, идентификатор и процессное время.
- 4) Полученный список процессов упорядочить по убыванию процессного времени.
- 5) Для п.4, показать 6 самых затратных процессов (максимальное процессное время).

Задание 1.6

- 1) Занести произвольную текстовую строку в переменную, имя которой начинается с символа V.
- 2) Создать директорию «№гг_Фамилия». (New-Item)

- 3) Вывести в файл (в созданной директории) список всех переменных, начинающихся с символа V. (dir с диска Variable:)
- 4) Вывести на экран содержимое созданного файла. (type)

Задание 1.7

Для следующих операции занести в файл:

```
$v_int = 150 + 401
```

```
$v_str = "Строка символов"
```

```
$v_log = ( 2 -eq 2 )
```

```
$v_obj = dir | Select-Object -First 5
```

- 1) Результат каждой операции.
- 2) После результата каждой операции записать тип результата через GetType().Name, а потом через Get-Member
- 3) Вывести на экран содержимое созданного файла.

Задание 1.8

- 1) В первую переменную поместить в одинарных кавычках (') строку, в которой помимо текста присутствуют две переменные.
- 2) Присвоить второй переменной числовое значение, а третьей – символьное значение.
- 3) Вывести на экран содержимое первой переменной.
- 4) Повторить п.1 с двойными кавычками (") и п.3.

Задание 1.9

- 1) Поместить строку из 4 символов в переменную \$var_1.
- 2) Повторить в переменной эту строку 3 раза.
- 3) Добавить в начало этой строки слово 'Старт', а в конец – слово 'Финиш'.
- 4) Вывести на экран содержимое созданной переменной.

Задание 1.10

Изменить функцию Hello, чтобы она запрашивала у пользователя его имя и вводила:
Добрый день, имя_пользователя !!!

Задание 1.11

Написать функцию:

- 1) Узнает имя пользователя и приветствует его.
- 2) Ввести числовое значения двух переменных.
- 3) Вывести на консоль результат деления первого числа на второе.

Задание 1.12

Написать функцию:

- 1) Узнает имя пользователя и приветствует его.
- 2) Ввести числовое значения двух переменных.
- 3) Вывести в файл строку:
число_1 / число_2 = результат_деления

Задание 1.13

Написать функцию:

- 1) Узнает имя пользователя и приветствует его.
- 2) Ввести числовое значения двух переменных.
- 3) Вывести в файл шесть строк показывающие все операции сравнения с этими числами:
число_1 оп_ср число_2 ->результат_сравнения

Например:
15 < 25 → True

Задание 1.14

- 1) Занести в переменную шаблон '?a*'
- 2) Сравнить с шаблоном следующие строки:
a, пустая строка, ba, ab, abc, babc, lavatar

Задание 1.15

- 1) Создать текстовый файл на диске с расширением TXT. В этот файл вывести содержимое корневого каталога диска C:
- 2) Сохранить в переменной \$ps_drive список дисков.
- 3) Записать в конец созданного в п.1 файла содержимое из переменной \$ps_drive
- 4) Вывести на экран содержимое созданного файла.

Задание 1.16

- 1) Создайте каталог с именем «№гр_Фамилия».
- 2) В созданном каталоге создайте несколько текстовых файлов и поместите в них несколько строк.
- 3) Сохраните содержимое созданного каталога в переменной \$ps_dir, а список дисков в переменной \$ps_drive.
- 4) Выведите на экран содержимое переменных \$ps_dir и \$ps_drive.

Задание 1.17

- 1) Вывести на экран список выполняемых процессов (Get-Process) по маске имени, в которое входит слово word (* word *).
- 2) Запустить программу Word: найти каталог, где расположен файл WinWord.exe и выполнить его.
- 3) Вывести на экран список выполняемых процессов (Get-Process) по маске имени, в которое входит слово word (* word *).
- 4) Остановить процесс WinWord (Stop-Process) по его идентификатору (id).

Задание 1.18

- 1) Вывести на экран список сервисов (Get-Service).
- 2) Посмотреть структуру объектов, описывающих сервисы (Get-Member).
- 3) Вывести информацию о 15 сервисах, упорядочив её по названию сервисов (Sort-Object). В выводимой информации отразить имя сервиса, статус, DisplayName (Select-Object).
- 4) Повторить п.3 с следующим ограничением: вывести на экран список сервисов, имена которых начинаются на букву 'w' (маска w*).

Задание 1.19

- 1) Вывести список всех выполняемых процессов. (Get-Process)
- 2) Вывести список всех выполняемых процессов, упорядочив его по убыванию процессорного времени. (Sort-Object)

Задание 1.20

- 1) Вывести список 10 процессов, которые максимально используют время центрального процессора. (Get-Process)
- 2) Занести этот список в переменную.(=)
- 3) Создать «№гр_Фамилия» директорию.(New-Item) В ней создать файл в который записать содержимое из переменной п.2.(>)

4) Вывести на экран содержимое созданного файла.(type)

1. Задания реконструктивного уровня

Задание 2.1

Написать функцию:

- 1) Ввести числовое значения трех переменных.
- 2) Вывести в файл три строки показывающие следующие операции с этими числами:

число_1 < число_2 и число_2 < число_3 = результат

число_1 < число_2 или число_2 < число_3 = результат

не (число_1 < число_3) = результат

Например:

15 < 25 и 25 < -5 = False

Задание 2.2

- 1) Вывести список всех сервисов (Get-Service).
- 2) Используя Where-Object вывести сервисы, которые остановлены.

Задание 2.3

- 1) Создать свой каталог с именем «№гр_Фамилия».
- 2) Найти каталог, в котором имеются и подкаталоги и файлы.
- 3) Вывести в файл в каталоге «№гр_Фамилия» список всех подкаталогов из п.2.(dir, >, where)
- 4) В конец этого файла добавить список всех файлов из п.2. (dir, >, where)
- 5) Содержимое созданного файла вывести на экран. (type)

Задание 2.4

- 1) Вывести в файл список всех процессов с положительным временем процессора (ps, >, where)
- 2) В конец этого файла добавить список процессов, которые более 10 сек. занимали процессор (ps, >, where)
- 3) Содержимое созданного файла вывести на экран. (type)

Задание 2.5

- 1) Вывести в файл список всех процессов с положительным временем процессора (ps, >, where)
- 2) В конец этого файла добавить список процессов, у которых в имени компании (Company) присутствует слово «Microsoft». По каждому процессу вывести информацию: имя, идентификатор, время старта, размер процессного времени, имя компании.
- 3) Содержимое созданного файла вывести на экран. (type)

2. Задачи творческого уровня

Задание 3_1

Создать функцию, которая:

- 1) запрашивает у пользователя путь в файловой системе и вводит его;
- 2) выводит зеленое сообщение, что такой путь существует
- 3) или выводит красное сообщение, что такой путь отсутствует.

Задание 3_2

Создать функцию, которая:

- 1) запрашивает у пользователя путь в файловой системе и вводит его;
- 2) выводит красное сообщение на белом фоне, что такой путь отсутствует;
- 3) выводит черное сообщение на желтом фоне, что путь указывает на папку;
- 4) выводит зеленое сообщение на желтом фоне, что путь указывает на файл.

Задание 3_3

Создать функцию, которая получает один параметр: путь в файловой системе. Если не указан параметр, то запросить путь у пользователя.

Выполняемые функцией действия:

- 1) выводит зеленое сообщение, что такой путь существует
- 2) или выводит красное сообщение, что такой путь отсутствует.

Задание 3_4

Создать функцию, которая получает один параметр: путь в файловой системе. Если не указан параметр, то запросить путь у пользователя.

Выполняемые функцией действия:

- 1) выводит красное сообщение на белом фоне, что такой путь отсутствует;
- 2) выводит черное сообщение на желтом фоне, что путь указывает на папку;
- 3) выводит зеленое сообщение на желтом фоне, что путь указывает на файл.

Задание 3_6

Создать функцию, которая выводит черное сообщение на желтом фоне о количестве остановленных сервисов (Get-Service).

Задание 3_7

Создать функцию, которая выводит зеленое сообщение на белом фоне о количестве работающих сервисов (Get-Service).

Задание 3_8

Создать функцию, которая выводит одно из ниже перечисленных зеленое сообщений (зеленое на белом фоне) о количестве сервисов (Get-Service) :

- 1) выполняемых сервисов больше остановленных на «число»
- 2) выполняемых сервисов меньше остановленных на «число»
- 3) выполняемых сервисов равно остановленным

Задание 3_9

Создать функцию, которая получает один параметр: букву. Если не указан параметр, то запросить значение у пользователя.

Функция выводит список всех процессов на заданную букву.

Задание 3_10

Создать функцию, которая получает один параметр: букву. Если не указан параметр, то запросить значение у пользователя.

Функция считает общее процессное время всех процессов на заданную букву. Вывести соответствующее сообщение желтым цветом на темно синем фоне.

Задание 3_11

Создать функцию, которая получает один параметр: букву. Если не указан параметр, то запросить значение у пользователя.

Функция считает количество всех процессов на заданную букву. Вывести соответствующее сообщение синим цветом на желтом фоне.

Задание 3_12

Создать функцию, которая получает один параметр: путь к директории в файловой системе. Если не указан параметр, то запросить путь у пользователя.

Функция считает количество файлов в директории и их общую длину и выводит одно из сообщений:

- 1) Отсутствует директория (белым на красном фоне)
- 2) В директории нет файлов (красным цветом на желтом фоне)
- 3) В директории XX файлов суммарным размером YY килобайт

Задание 3_14

Создать функцию, которая получает один параметр: букву. Если не указан параметр, то запросить значение у пользователя.

Функция считает общий объем памяти, необходимый для всех процессов на заданную букву и у которых процессное время положительно. Вывести соответствующее сообщение желтым цветом на темно синем фоне.

Задание 3_15

Создать функцию, которая получает один параметр: число N. Если не указан параметр, то запросить значение у пользователя.

Функция считает общий объем памяти, необходимый для N процессов с максимальным процессным временем. Вывести соответствующее сообщение желтым цветом на темно синем фоне.

Задание 3_16

Создать функцию, которая получает один параметр: имя массива целых чисел.

Подсчитывает сумму элементов, удовлетворяющих условию:

- a) значения кратны 5;
- b) нечетные отрицательны значения;

Вывести результат желтым цветом на темно синем фоне.

Задание 3_17

Создать функцию, которая получает один параметр: имя массива целых чисел.

Подсчитать сумму и количество элементов, удовлетворяющих условию:

- a) значения кратны 5;
- b) нечетные отрицательны значения;

Вывести результат желтым цветом на темно синем фоне.

Задание 3_18

Создать функцию, которая получает один параметр: имя массива чисел.

Выполнить преобразование массива:

отрицательные значения возвести в квадрат, а у положительных и четных \square изменить знак числа.

Полученный массив вывести на экран.

Задание 3_19

Создать функцию, которая получает один параметр: имя массива целых чисел.

Создать результирующий массив из элементов исходного массива, которые удовлетворяют условию:

- a) кратны 3 и не кратны 5;

b) являются нечетными числами и имеющие четные номера в массиве.
Исходный и полученный массивы и количество их элементов вывести на экран.

Задание 3_20

Создать функцию, которая получает один параметр: имя массива целых чисел.
Ввести контрольное число X. Подсчитать сумму и количество всех элементов массива, следующих за первым элементов, значение которого равно X. Если таких элементов нет, то выдать соответствующее сообщение.

Задание 3_21

Создать функцию, которая получает один параметр: имя массива чисел.
Подсчитать сумму и количество элементов массива, значения которых меньше среднего арифметического всех элементов исходного массива.
Результат вывести на экран.

Задание 3_22

Создать функцию, которая получает один параметр: имя файла. Если не указан параметр, то запросить имя файла у пользователя.

Выполняемые функцией действия:

- 1) Вывести пользователю на консоль список дисков.
- 2) Запросить у пользователя имя диска.
- 3) Проверить на существование диска.
- 4) В указанный файл вывести содержимое корневого каталога заданного диска.
- 5) Записать в конец файла список дисков.
- 6) Вывести на экран содержимое созданного файла.

Задание 3_22a

- 1) Найти директорию, в которой имеются и подкаталоги и файлы. Например, директория с программами MS Office.
- 2) Перейти в эту директорию. (cd)
- 3) Показать на экране содержимое этой директории. (dir)
- 4) Вывести на экран только подкаталоги. Использовать Where-Object с условием PSIsContainer.
- 5) Вывести на экран только файлы. Использовать Where-Object с условием PSIsContainer.
- 6) Создать каталог с именем PowerShell (New-Item).
- 7) В каталог PowerShell записать файл с содержимым из п.4 и п.5. (>)
- 8) Просчитать (Measure-Object) количество и суммарный размер файлов из п.5, а так же минимальный и максимальный размеры. И дозаписать в конец файла.>>>

Задание 3_23

Используя оператор Measure подсчитать количество остановленных сервисов (Get-Service) и вывести соответствующее сообщение синим сообщением на желтом фоне (Write-Host) .

Задание 3_24

Создать массив целых чисел.
Используя оператор Measure подсчитать сумму, количество элементов массива и среднего арифметического всех элементов исходного массива.

Задание 3_25

- 1) Найти директорию, в которой находится много различных файлов. Например, директория программ MS Office:
`dir c:\program* -Recurse -Filter *excel*.exe`
- 2) Перейти в эту директорию. (`cd`)
- 3) Показать на экране содержимое этой директории. (`dir`)
- 4) Вывести на экран сгруппированные объекты текущей директории по свойству `Attributes`.

Задание 3_26

- 1) Вывести на экран сервисы сгруппированные по свойству `Status`. (`Get-Service`)
- 2) Вывести на экран сервисы сгруппированные по свойству `Status` подавив вывод перечисления элементов в группах (`NoElement`).

Задание 3_27

- 1) Вывести на экран группировку файлов директории с программами MS Office по их расширению (свойство `Extension`).
- 2) Повторить п.1, но подавив вывод перечисления элементов групп (`NoElement`).
- 3) Повторить п.1, но вывести только группы, у которых более одного элемента (`$.Count`). Использовать командлет `Where-Object {...}`

Задание 3_28

- 1) Вывести на экран все процессы выполняемые в системе (`Get-Process`).
- 2) Повторить п.1, упорядочив их по процессному времени (`CPU`). Командлет `Sort-Object`.
- 3) Повторить п.2, но упорядочив по убыванию процессного времени (параметр-переключатель `-Descending`).
- 4) Повторить п.3, но для процессов вывести только идентификатор (`id`), имя (`ProcessName`) и название компании разработчика (`Company`). Командлет `Select-Object`.
- 5) Повторить п.4, сгруппировав процессы по названию компаний разработчиков (`Company`). Командлет `Group-Object`.
- 6) Повторить п.5, но вывести только группы, у которых более одного элемента (`$.Count`). Использовать командлет `Where-Object`.
- 7) Повторить п.6, не выводя группы, у которых пустое имя (`$.Name`). Составить сложное логическое выражение, используя логический оператор И (`-and`).

Задание 3_30

Вывести на экран текущий год.

Задание 3_31

Написать функцию определения сколько дней прошло с начала года до первого числа текущего месяца.

Задание 3_32

Написать функцию определения сколько дней осталось до конца текущего года.

Задание 3_33

Написать функцию `Age`, которая спрашивает у пользователя имя и год рождения и выдает сообщение (черным цветом на желтом фоне):
«Имя, Ваш возраст NN»

Задание 3_34

Изменить функцию Age, проверив корректность ввода года рождения (не больше текущего)

Задание 3_35

Написать функцию на вход которой передаются 3 параметра (год, месяц, день), а в результате выдает номер дня недели для этой даты.

Задание 3_36

Написать функцию, которая спрашивает у пользователя год и выводит (черным цветом на желтом фоне) на консоль список установленного на данный компьютер в указанном году ПО (Get-WmiObject win32_product)

Задание 3_37

Написать функцию, которая спрашивает у пользователя год (ПГОД) и выводит в файл список установленного на данный компьютер ПО (Get-WmiObject win32_product) по годам (где ПЕРЕДПГОД= ПГОД-1). Вид выводимого списка:

Установлено в текущем ПГОД году

...

...

Установлено в предыдущем ПЕРЕДПГОД году

...

...

Установлено до ПЕРЕДПГОД года

...

...

Вопросы для опроса:

1. С какими объектами взаимодействует операционная система?
2. Назовите основные функции ОС
3. Место ОС в программном обеспечении компьютеров.
4. Эволюция ОС.
5. Классификация ОС.
6. Интерфейсы операционных систем.
7. Эффективность ОС.
8. Пользовательский и привилегированный режимы работы
9. Функции, включаемые в ядро операционной системы
10. Назовите возможные состояния процесса
11. Дайте определение виртуальной памяти. Перечислите варианты организации такой памяти.
12. Основные компоненты: драйверы, файловая система, система прерываний. Дайте им характеристику.
13. Концепция процессов и потоков. Роль процессов, потоков и волокон в мультизадачности.
14. Иерархическая организация памяти.
15. Распределение памяти динамическими разделами.
16. Распределение памяти перемещаемыми разделами.

17. Виртуальная память.
18. Преобразование виртуальных адресов в физические.
19. Задачи ОС по управлению файлами и устройствами.
20. Структура файловой системы.
21. Обработчики прерываний и драйверы устройств.
22. Организация параллельной работы устройств ввода-вывода и процессора.
23. Согласование скоростей обмена и кэширование данных.
24. Обеспечение логического интерфейса между устройствами и остальной частью системы.
25. Угрозы информационной безопасности ОС.
26. Управление доступом к ресурсам.
27. Формальные модели систем безопасности.

Критерии оценки (в баллах):

- 16-20 баллов выставляется студенту, если он знает верно и в полном объеме: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем и **умеет верно и в полном объеме:** осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- 11-15 баллов выставляется студенту, если он знает с незначительными замечаниями:

современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем и **умеет с незначительными замечаниями:** осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- 6-10 баллов выставляется студенту, если он знает на базовом уровне, с ошибками: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем и **умеет на базовом уровне, с ошибками:** осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- 0-5 баллов выставляется студенту, если он не знает на базовом уровне: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом

построения операционных систем и **не умеет на базовом уровне:** осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

Задания для текущего контроля

Ролевая игра

Индикаторы достижения:ПК-5.2

Название: Имитация Управление процессами

Роли для каждого Процесса (количество процессов = 5)

Роль Таймера

Роль Очередь ожидающих завершения ввода-вывода

Роль Диспетчер

Роль Аналитик

Роль таблица процессов:

Роль CPU (Процессор):

Ситуации:

1. Стартовал новый процесс

2. Завершился вычислительный шаг процесса (id, шаг, длительность)

3. Завершился шаг вв-выв процесса (id, шаг, длительность)

Завершение игры: когда все процессы выполнены

Критерии оценки (в баллах):

- **5 баллов** выставляется студенту, если он умеет верно и в полном объеме: осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- **3-4 баллов** выставляется студенту, если он умеет с незначительными замечаниями: осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- **1-2 баллов** выставляется студенту, если он умеет на базовом уровне, с ошибками: осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- **0 баллов** выставляется студенту, если он не умеет на базовом уровне: осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

Расчетно-аналитические задания

Индикаторы достижения:ПК-5.2

Задача № 1

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге количество файлов, размер которых (Length) более 1000 байт.
2. Поместить в переменную \$VR подсчитанное количество файлов и список файлов, упорядоченный по убыванию длины файла.
3. Вывести на экран содержимое переменной \$VR.

Задача № 2

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге количество файлов.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество файлов и список из 15 файлов, упорядоченный по убыванию даты создания (CreationTime).
3. Вывести на экран содержимое файла *f_Фамилия*.

Задача № 4

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге количество файлов, размер которых (Length) более 1000 байт.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество файлов и список файлов, упорядоченный по убыванию длины файла.
3. Вывести на экран содержимое файла *f_Фамилия*.

Задача № 5

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге количество файлов.
2. Поместить в переменную \$VR подсчитанное количество файлов и список из 15 файлов, упорядоченный по убыванию даты создания (CreationTime).
3. Вывести на экран содержимое переменной \$VR.

Задача № 7

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву.
2. Поместить в переменную \$VR подсчитанное количество процессов и список этих процессов, упорядоченный по убыванию процессного времени (CPU).

3. Вывести на экран содержимое переменной \$VR.

Задача № 8

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество процессов и список этих процессов, упорядоченный по убыванию процессного времени (CPU).
3. Вывести на экран содержимое файла *f_Фамилия*.

Задача № 9

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву.
2. Поместить в переменную \$VR подсчитанное количество процессов и список из 15 этих процессов, упорядоченный по убыванию процессного времени (CPU).
3. Вывести на экран содержимое переменной \$VR.

Задача № 10

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество процессов и список из 15 этих процессов, упорядоченный по убыванию процессного времени (CPU).
3. Вывести на экран содержимое файла *f_Фамилия*.

Задача № 11

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге (PSisContainer) количество директорий.
2. Поместить в переменную \$VR подсчитанное количество директорий и список директорий, упорядоченный по убыванию их имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 12

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге (PSisContainer) количество директорий.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество директорий и список директорий, упорядоченный по убыванию их имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 13

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге(PSisContainer)количество директорий.
2. Поместить в переменную \$VRподсчитанное количество директорийи список из 15 директорий, упорядоченный по убыванию их имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 14

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге(PSisContainer)количество директорий.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество из 15 директорийи список директорий, упорядоченный по убыванию их имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 15

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге(PSisContainer)количество файлов и директорий.
2. Поместить в переменную \$VRподсчитанное количество и два списка файлов и директорий, причем каждый упорядочен по имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 16

Создать функцию, которая получает один параметр: путь к каталогу

1. Подсчитатьв указанном каталоге(PSisContainer)количество файлов и директорий.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество и два списка файлов и директорий, причем каждый упорядочен по имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 17

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитатьв указанном каталоге(PSisContainer)количество файлов и директорий.
2. Поместить в переменную \$VRподсчитанное количество и два списка из 15 файлов и из 10 директорий, причем каждый упорядочен по имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 18

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге(PSisContainer)количество файлов и директорий.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество и два списка из 15 файлов и из 10 директорий, причем каждый упорядочен по имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 19

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать в указанном каталоге(PSisContainer)количество файлов, их общую длину и максимальную длину.
2. Поместить в переменную \$VRподсчитанное полученные результаты и список файлов, упорядоченный по убыванию их имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 20

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге(PSisContainer)количество файлов, их общую длину и максимальную длину.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное полученные результаты и список файлов, упорядоченный по убыванию их имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 21

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге(PSisContainer)количество файлов, их общую длину и максимальную длину.
2. Поместить в переменную \$VRподсчитанное полученные результаты и список из 15 файлов, упорядоченный по убыванию их имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 22

Создать функцию, которая получает один параметр: путь к каталогу.

1. Подсчитать в указанном каталоге(PSisContainer)количество файлов, их общую длину и максимальную длину.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное полученные результаты и списки из 15 файлов, упорядоченный по убыванию их имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 23

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество переменных PowerShell на заданную букву (диск Variable).
2. Поместить в переменную \$VR подсчитанное количество переменных PowerShell и список этих переменных, упорядоченный по имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 24

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество переменных PowerShell на заданную букву (диск Variable).
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество переменных PowerShell и список этих переменных, упорядоченный по имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 25

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество алиасов на заданную букву (диск Alias).
2. Поместить в переменную \$VR подсчитанное количество алиасов и список этих алиасов, упорядоченный по имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 26

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество алиасов на заданную букву (диск Alias).
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество алиасов и список этих алиасов, упорядоченный по имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 27

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество переменных Windows на заданную букву (диск Env).
2. Поместить в переменную \$VR подсчитанное количество переменных Windows и список этих переменных, упорядоченный по имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 28

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество переменных Windows на заданную букву (диск Env).
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество переменных Windows и список этих переменных, упорядоченный по имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 29

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество функций на заданную букву (диск Function).
2. Поместить в переменную \$VR подсчитанное количество функций и список этих функций, упорядоченный по имени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 30

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество функций на заданную букву (диск Function).
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество функций и список этих функций, упорядоченный по имени.
3. Выводит на экран содержимое файла *f_Фамилия*.

Задача № 31

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву, у которых процессорное время положительное (больше нуля) (CPU)..
2. Поместить в переменную \$VR подсчитанное количество процессов и список этих процессов, упорядоченный по убыванию процессного времени
3. Вывести на экран содержимое переменной \$VR.

Задача № 32

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву, у которых процессорное время положительное (больше нуля) (CPU).
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество процессов и список этих процессов, упорядоченный по убыванию процессного времени.
3. Вывести на экран содержимое файла *f_Фамилия*.

Задача № 33

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву, у которых процессорное время положительное (больше нуля)(CPU).
2. Поместить в переменную \$VR подсчитанное количество процессов и список из 15 этих процессов, упорядоченный по убыванию процессного времени.
3. Вывести на экран содержимое переменной \$VR.

Задача № 34

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву, у которых процессорное время положительное (больше нуля)(CPU).
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество процессов и список из 15 этих процессов, упорядоченный по убыванию процессного времени.
3. Вывести на экран содержимое файла *f_Фамилия*.

Задача № 35

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву, у которых процессорное время положительное (больше нуля)(CPU), но меньше 1000.
2. Поместить в переменную \$VR подсчитанное количество процессов и список этих процессов, упорядоченный по убыванию процессного времени
3. Вывести на экран содержимое переменной \$VR.

Задача № 36

Создать функцию, которая получает один параметр: заданная буква.

1. Подсчитать количество процессов на заданную букву, у которых процессорное время положительное (больше нуля)(CPU), но меньше 1000.
2. Поместить в файл (имя файла: *f_Фамилия*) подсчитанное количество процессов и список этих процессов, упорядоченный по убыванию процессного времени.
3. Вывести на экран содержимое файла *f_Фамилия*.

Критерии оценки (в баллах):

- **11-15 баллов** выставляется студенту, если он знает верно и в полном объеме: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем и умеет верно и в полном объеме: осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать

вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- 6-10 баллов выставляется студенту, если он знает с незначительными замечаниями:

современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем и **умеет с незначительными замечаниями:** осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- 1-5 баллов выставляется студенту, если он знает на базовом уровне, с ошибками: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем и **умеет на базовом уровне, с ошибками:** осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

- 0 баллов выставляется студенту, если он не знает на базовом уровне: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем и **не умеет на базовом уровне:** осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.

Задания для творческого рейтинга

Тематика докладов

Индикаторы достижения:ОПК-5.2

- 01 Обзор мобильных ОС: Android (Google), iOS (Apple), MiUi (Xiaomi), Bada (Samsung), KaiOS (TechnologiesInc., Google), FireOS (Amazon), Tizen (Intel, Samsung), WindowsMobile (Microsoft), Symbian (Nokia), SailfishOS (Jolla), PalmOS (RIM), BlackBerryOS, Fuchsia (Google)
- 02 Гипервизоры первого и второго типов
- 03 Технология работы виртуальной машины
- 04 Основные модели облачных технологий (IaaS, PaaS, SaaS)
- 05 Статистика использования ОС на персональных компьютерах (вкл. ноутбуки) в разрезе годов и географии
- 06 Статистика использования ОС на мобильных гаджетах в разрезе годов и географии
- 07 Статистика использования серверных ОС в разрезе годов и географии

- 08 История развития Windows
- 09 История развития Unix и Unix-подобных ОС
- 10 Технологии организации Удаленного рабочего стола
- 11 ОС на базе Linux: перечень и основные характеристики
- 12 Архитектура ОС Android
- 13 Процесс загрузки ОС Android
- 14 Архитектура ОС Linux
- 15 Архитектура ОС iOS (Apple)
- 16 Архитектура ОС MacOS (Apple)
- 17 Архитектура ОС MiUi (Xiaomi)
- 18 Архитектура ОС Chrome OS (Google)
- 19 Архитектура ОС Аврора (Mobile OS RUS)
- 20 Архитектура ОС KasperskyOS
- 21 Возможности Проводника (Windows)
- 22 Возможности Менеджера (диспетчер) файлов (Linux)
- 23 Возможности Диспетчера задач (Windows) (вызов Ctrl+Shft+ESC)
- 24 Способы создания ярлыка для запуска приложения (Windows)
- 25 Active Directory (Windows)
- 26 Сравнительный анализ возможностей PowerShell(Windows) и Bash(Linux)
- 27 UEFI (Unified Extensible Firmware Interface), описание разделов PowerShell (PS): описание, пример использования, 2 задачи для аудиторрии.
- 28 PS: Преобразование типов данных (int↔string) и использование статических методов (математические методы Sqrt, Pow, Exp, Log и т.п.)
- 29 PS: установка PROMPT
- 30 PS: командлета Format-Table
- 31 PS: командлеты Push-Location и Pop-Location
- 32 PS: Возможности командлеты Get-WmiObject (WindowsManagementInstrumentation)
PowerShell (PS): Свойства описывающие Процессы (Get-Process): назначение свойства, тип значения, возможные значения, примеры использования, 2 задачи для аудиторрии.
- 33 Свойства: Id, MachineName, ProcessName, SessionId, Site, Container, ExitCode, ExitTime, HasExited
- 34 Свойства: PagedMemorySize, PagedSystemMemorySize, PagedMemorySize64, PagedSystemMemorySize64, PeakPagedMemorySize, PeakVirtualMemorySize, PeakWorkingSet, VirtualMemorySize, PrivateMemorySize
- 35 Свойства: Handle, HandleCount, SafeHandle, MainWindowHandle
- 36 Свойства: TotalProcessorTime, UserProcessorTime, CPU, StartTime, Company, Description, Product
- 37 Свойства: MainModule, Modules, MainWindowTitle
- 38 Свойства описывающие потоки внутри процесса (Threads): Id, ThreadState, WaitReason, StartAddress, BasePriority, CurrentPriority, TotalProcessorTime, UserProcessorTime PowerShell (PS): Свойства описывающие Директории (dir): назначение свойства, тип значения, возможные значения, примеры использования, 2 задачи для аудиторрии.
- 39 Свойства: Name, FullName, Parent, Root, Extension, BaseName, Target PSDrive
- 40 Свойства: Mode, Attributes, PSIsContainer, PSChildName, Exists PSParentPath
- 41 Свойства: CreationTimeCreationTimeUtcLastAccessTimeLastWriteTimeLinkType

PowerShell (PS): Свойства описывающие Файлы (dir): назначение свойства, тип значения, возможные значения, примеры использования, 2 задачи для аудитории.

42 Свойства: Name, FullName, Parent, Root, DirectoryName, IsReadOnly, Length

Критерии оценки (в баллах):

- **16-20 баллов** выставляется студенту, если он знает верно и в полном объеме: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.

- **11-15 баллов** выставляется студенту, если он знает с незначительными замечаниями:

современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.

- **6-10 баллов** выставляется студенту, если он знает на базовом уровне, с ошибками: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.

- **0-5 баллов** выставляется студенту, если он не знает на базовом уровне: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ХАРАКТЕРИЗУЮЩИЕ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ ВО ВРЕМЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Структура зачетного задания

<i>Наименование оценочного средства</i>	<i>Максимальное количество баллов</i>
<i>Теоретический вопрос из лекционного материала</i>	<i>15</i>
<i>Теоретический вопрос из лекционного материала</i>	<i>15</i>
<i>Практическое задание</i>	<i>10</i>

Задания, включаемые в зачетное задание

Перечень вопросов к зачету с оценкой:

1. Определения операционной системы
2. Место операционной системы в компьютерной системе
3. Назначение операционной системы
4. Функции операционной системы
5. Схема работы механизма прерывания
6. Шаги обработки прерывания
7. Два основных класса прерывания
8. Определение ресурса в операционной системе
9. Основные виды ресурсов
10. Основные задачи операционной системы
11. Эволюция операционных систем
12. Классификация операционных систем
13. Работа виртуальной машины
14. Гипервизоры первого и второго типов
15. Основные модели облачных технологий
16. Архитектура операционных систем
17. Пользовательский и привилегированный режимы работы
18. Функции, включаемые в ядро операционной системы
19. Иерархия слоев операционной системы
20. Монолитная и микроядерная архитектуры
21. Понятие процесса в операционной системе
22. Понятие мультизадачности в операционной системе
23. Объекты, входящие в состав процесса
24. Назначение Блока управления процессом (PCB)
25. Назначение Таблицы процессов
26. Содержимое Блока управления процессом (PCB)
27. Фоновые процессы
28. Причины создание процесса
29. Назначение диспетчера процессов
30. Возможные состояния процесса
31. Условия перехода процесса из одного состояния в другое
32. Назначение планирования процессов в операционной системе
33. Методы планирования процессов в операционной системе

34. FIFO метод планирования процессов
35. Метод планирования процессов «Кратчайшая работа следующей»
36. Метод планирования процессов Round-robin (циклический)
37. Метод планирования процессов «Многоуровневая очередь»
38. Метод планирования процессов «Многоуровневая очередь с обратной связью»
39. Физическое представление процесса
40. Этапы создания процесса
41. Отличие процессов от потоков
42. Функции ОС по управлению памятью
43. Концепция иерархии памяти
44. Адресное пространство процесса
45. Свопинг
46. Виртуальная память
47. Страничный способ организации виртуальной памяти
48. Задачи, решаемые поддержкой виртуального адресного пространства
49. Таблица страниц
50. Трансляция виртуального адреса в физический
51. Страничное прерывание
52. Управление свободной памятью
53. Этапы преобразования символических имен в адрес физической памяти
54. Определение файла
55. Назначение файловой системы
56. Состав файловой системы
57. Полное имя файла
58. Расширение имени файла
59. Типы файлов
60. Атрибуты файлов
61. Каталоги файлов
62. Технология непрерывного размещения файлов
63. Технология размещения файлов с использованием связанного списка
64. Технология размещения файлов с помощью связанного списка, использующего таблицу в памяти
65. Технология размещения файлов с использованием i-узлов
66. Операции с файлами
67. Операции с каталогами файлов
68. Блочные и символьные устройства

69. Контроллеры устройств
70. Прямой доступ к памяти (DMA)
71. Основные требования к ОС при обеспечении процесса ввода-вывода
72. Четыре слоя программного обеспечения ввода-вывода
73. Драйвер устройства
74. Независимый от устройств слой операционной системы

Практические задания для зачета:

Задание 1 для реализации в среде PowerShell

- 1) Создайте каталог с именем «№ гр_Фамилия».
- 2) В созданном каталоге создайте несколько текстовых файлов и поместите в них несколько строк.
- 3) Сохраните содержимое созданного каталога в переменной `$ps_dir`, а список дисков в переменной `$ps_drive`.
- 4) Выведите на экран содержимое переменных `$ps_dir` и `$ps_drive`.

Задание 2 для реализации в среде PowerShell

- 1) Вывести в файл список всех процессов с положительным временем процессора (`ps, >, where`)
- 2) В конец этого файла добавить список процессов, которые более 10 сек. занимали процессор (`ps, >, where`)
- 3) Содержимое созданного файла вывести на экран. (`type`)

Задание 3 для реализации в среде PowerShell

Создать функцию, которая получает один параметр: букву. Если не указан параметр, то запросить значение у пользователя.

Функция считает общий объем памяти, необходимый для всех процессов на заданную букву и у которых процессное время положительно. Вывести соответствующее сообщение желтым цветом на темно синем фоне.

Показатели и критерии оценивания планируемых результатов освоения компетенций и результатов обучения, шкала оценивания

Шкала оценивания		Формируемые компетенции	Индикатор достижения компетенции	Критерии оценивания	Уровень освоения компетенций
85 – 100 балло в	«зачтено / отлично»	ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2. Осуществляет установку и конфигурирование операционных систем	<p>Знает верно и в полном объеме: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.</p> <p>Умеет верно и в полном объеме: осуществлять установку и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.</p>	Продвинутый
70 – 84 балло в	«зачтено / хорошо»	ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2. Осуществляет установку и конфигурирование операционных систем	<p>Знает с незначительными замечаниями: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды;</p>	Повышенный

				<p>объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.</p> <p>Умеет с незначительными замечаниями: осуществлять установку и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.</p>	
50 – 69 баллов	«зачтено / удовлетворительно»	ОПК-5. Способен установить программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2. Осуществляет установку и конфигурирование операционных систем	<p>Знает на базовом уровне, с ошибками: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.</p> <p>Умеет на базовом уровне, с ошибками: осуществлять</p>	Базовый

				<p>инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.</p>	
<p>менее 50 баллов</p>	<p>«незачтено / неудовлетворительно»</p>	<p>ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем</p>	<p>ОПК-5.2. Осуществляет инсталляцию и конфигурирование операционных систем</p>	<p>Не знает на базовом уровне: современные операционные системы и оболочки, архитектуру и интерфейсы операционных систем, операционные оболочки и среды; объекты мультипрограммного вычислительного процесса, концепцию управления мультипрограммным вычислительным процессом построения операционных систем.</p> <p>Не умеет на базовом уровне: осуществлять инсталляцию и конфигурирование операционных систем, управлять мультипрограммным вычислительным процессом; создавать и настраивать вычислительную среду для реализации бизнес-процессов в корпоративных сетях предприятий.</p>	<p>Компетенции не сформированы</p>