

Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 26.06.2026 11:09:14	МИНОБНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)	
Уникальный идентификатор документа: 04c19ed8b1b981958e57648589ab78888922529	Фонд оценочных средств для промежуточной аттестации по дисциплине "Операционные системы" по направлению подготовки (специальности) "09.03.04 Программная инженерия" направленности (профилю) Разработка программно-информационных систем ФГБОУ ВО «ЧелГУ»	стр. 1

**Фонд оценочных средств для промежуточной аттестации по дисциплине
Операционные системы**

Направление подготовки (специальность)

09.03.04 Программная инженерия

Направленность (профиль)

Разработка программно-информационных систем

Присваиваемая квалификация (степень)

бакалавр

Форма обучения

очно-заочная форма обучения

Год(ы) набора 2026

Челябинск 2026 г.

**09.03.04 Программная инженерия профиль Разработка программно-информационных систем,
дисциплина Операционные системы, 2026 год набора, очно-заочная форма обучения**

Фонд оценочных средств дисциплины (модуля) одобрен и рекомендован:

Проректор по учебной работе утверждено 27.02.2026 А.А. Саламатов

Ученым советом института информационных технологий

Протокол заседания № 7 от 26.02.2026

Председатель Ученого совета
института информационных
технологий

согласовано

Ю.В. Петриченко

Заседанием кафедры информационных технологий и экономической информатики

Протокол заседания №7 от 26.02.2026

Заведующий кафедрой

согласовано

С.А. Скрипов

Автор (составитель)

С.А. Скрипов

**Структура фондов оценочных средств соответствует приказу ректора ФГБОУ ВО «ЧелГУ» от 27
сентября 2022 № 573-1**



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	6
3.1. Виды оценочных средств	6
3.2. Содержание оценочных средств	8
4. Порядок проведения и критерии оценивания промежуточной аттестации	41
4.1. Порядок проведения промежуточной аттестации	41
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	41
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций	41



1. Паспорт фонда оценочных средств

Направление подготовки: 09.03.04 Программная инженерия

Направленность: Разработка программно-информационных систем

Дисциплина: Операционные системы.

Семестр (семестры) изучения: 3, 4

Форма промежуточной аттестации: экзамены

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Операционные системы» направлено на формирование компетенций, приведённых в 1.

Таблица 1. Результаты обучения по дисциплине.

Коды компетенции и согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решения задач профессиональной деятельности;	ОПК-2.1. Демонстрирует знание инструментальных средств, готового программного обеспечения и библиотек; знаком с содержанием Единого реестра российских программ ОПК-2.2. Умеет выбирать и использовать инструментальные средства, готовое программное обеспечение и библиотеки ОПК-2.3. Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения и сетевых коммуникаций	Знать: методы использования инструментальных средств для автоматизации работы с операционной оболочкой; программное обеспечение и библиотеки для системного программирования Уметь: выбирать и использовать инструментальные средства, готовое программное обеспечение и библиотеки для работы с оболочкой и системного программирования Владеть: навыками решения задач анализа, интеграции программного обеспечения для различных операционных систем
ОПК-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем;	ОПК-5.1. Обладает базовыми знаниями основ устройства и администрирования программного и аппаратного обеспечения информационных систем и баз данных ОПК-5.2. Демонстрирует умения устанавливать программное обеспечение информационных систем и баз данных ОПК-5.3. Имеет практический опыт установки и сопровождения программного и аппаратного обеспечения информационных систем и баз данных	Знать: основы устройства и администрирования операционных систем Уметь: устанавливать программное обеспечение с использованием возможностей операционной оболочки Владеть: навыками установки программного обеспечения из репозитория



ПК-1	Владение навыками использования операционных систем, сетевых технологий, современных языков программирования, технологий обработки данных, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных	ПК-1.1. Демонстрирует знание основ операционных систем, сетевых технологий, языков программирования, баз данных и технологий обработки данных, основ проектирования интерфейсов, языков и методов формальных спецификаций ПК-1.2. Демонстрирует умения разрабатывать системное и прикладное программное обеспечение с использованием языков и технологий программирования, баз данных, сетевых технологий и операционных систем, языков и методов формальных спецификаций ПК-1.3. Имеет практический опыт использования операционных систем, современных языков программирования, систем управления базами данных и технологий обработки данных, средств разработки программного интерфейса	Знать: основы операционных систем, языков программирования, основ проектирования интерфейсов, языков и методов формальных спецификаций Уметь: разрабатывать системное и прикладное программное обеспечение с использованием языков и технологий программирования, операционных систем, языков и методов формальных спецификаций Владеть: навыками использования операционных систем, современных языков программирования, средств разработки программного интерфейса
------	---	---	--



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

№ п/п	Код компетенции/ планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1	ОПК-2.1. Демонстрирует знание инструментальных средств, готового программного обеспечения и библиотек; знаком с содержанием Единого реестра российских программ Знать: основы операционных систем, языков программирования, основ проектирования интерфейсов, языков и методов формальных спецификаций	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 12-23, 27-52, 63-69
2	ОПК-2.2. Умеет выбирать и использовать инструментальные средства, готовое программное обеспечение и библиотеки Уметь: разрабатывать системное и прикладное программное обеспечение с использованием языков и технологий программирования, операционных систем, языков и методов формальных спецификаций	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 1-11, 24-26, 63-69, 70-184
3	ОПК-2.3. Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения и сетевых коммуникаций Владеть: навыками использования операционных систем, современных языков программирования, средств разработки программного интерфейса	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 1-11, 53-62
4	ОПК-5.1. Обладает базовыми знаниями основ устройства и	Понятие, история классификация ОС. Управление памятью.	Тест	Задания теста № 12-23, 27-52, 63-69



	администрирования программного и аппаратного обеспечения информационных систем и баз данных Знать: основы устройства и администрирования операционных систем	Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды		
5	ОПК-5.2. Демонстрирует умения устанавливать программное обеспечение информационных систем и баз данных Уметь: устанавливать программное обеспечение с использованием возможностей операционной оболочки	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 1-11, 24-26, 63-69, 70-184
6	ОПК-5.3. Имеет практический опыт установки и сопровождения программного и аппаратного обеспечения информационных систем и баз данных Владеть: навыками установки программного обеспечения из репозитория	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 1-11, 53-62
7	ПК-1.1. Демонстрирует знание основ операционных систем, сетевых технологий, языков программирования, баз данных и технологий обработки данных, основ проектирования интерфейсов, языков и методов формальных спецификаций Знать: методы использования инструментальных средств для автоматизации работы с операционной оболочкой; программное обеспечение и библиотеки для системного программирования	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 12-23, 27-52, 63-69
8	ПК-1.2. Демонстрирует умения разрабатывать системное и прикладное программное обеспечение с использованием языков и технологий программирования, баз данных, сетевых технологий и операционных систем, языков и методов	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 1-11, 24-26, 63-69, 70-184



	формальных спецификаций Уметь: выбирать и использовать инструментальные средства, готовое программное обеспечение и библиотеки для работы с оболочкой и системного программирования			
9	ПК-1.3. Имеет практический опыт использования операционных систем, современных языков программирования, систем управления базами данных и технологий обработки данных, средств разработки программного интерфейса Владеть: навыками решения задач анализа, интеграции программного обеспечения для различных операционных систем	Понятие, история классификация ОС. Управление памятью. Файловые системы. Мультипрограммирование Операционные оболочки Архитектура Windows Операционные среды	Тест	Задания теста № 1-11, 53-62

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2. Содержание оценочных средств

База тестовых вопросов для 3 семестра

№ п/п	Формулировка вопроса	Варианты ответов (полужирным шрифтом – верные варианты)
1.	Команда <code>echo off</code>	a. Отключит вывод команд на экран во время выполнения b. Выключит режим раннего связывания для переменных окружения c. Позволяет выводить данные не на экран, а в файл
2.	В командной строке Windows команда <code>dir</code> позволяет:	a. Получить список файлов и подкаталогов b. Изменить атрибуты файлов и каталогов c. Переименовать файлы и каталоги d. Переименовать только файлы (но не каталоги)
3.	В командной строке Windows команда <code>md</code> позволяет:	a. Получить список файлов



		<p>и подкаталогов b. Изменить атрибуты файлов и каталогов c. Удалить файл d. Создать каталог</p>
4.	Для получения первого параметра, переданного в командный файл Windows (.bat) необходимо использовать следующее выражение:	<p>a. %1 b. %0 c. %~x1 d. \$PARAM[1] e. \$PARAM[0]</p>
5.	В командном файле (.bat) оператор > используется:	<p>a. Для перенаправления вывода в файл с перезаписью указанного файла b. Для перенаправления вывода в файл с записью в конец указанного файла c. Для извлечения параметров, переданных в командной строке</p>
6.	В командном файле (.bat) оператор >> используется:	<p>a. Для перенаправления вывода в файл с перезаписью указанного файла b. Для перенаправления вывода в файл с записью в конец указанного файла c. Для извлечения параметров, переданных в командной строке</p>
7.	Для использования связывания времени выполнения (то есть чтобы в составных выражениях подставлялись актуальные значения переменных окружения) в командном файле Windows необходимо использовать следующий синтаксис:	<p>a. !k! для переменной k b. %k% для переменной k c. &k& для переменной k d. ?k? для переменной k</p>
8.	Следующий сценарий: SET /a k=0 FOR /R %~i IN (*) do set /a k=k+1 echo %k%	<p>a. Подсчитает количество файлов в текущем каталоге, включая файлы в подкаталогах b. Выведет на экран все имена файлов в текущем каталоге, не включая подкаталоги c. Выведет на экран все имена файлов в текущем каталоге, включая подкаталоги d. Выведет на экран все файлы, в именах которых есть "i"</p>
9.	Следующая команда в сценарии: dir /B *.doc>>1.txt	<p>a. Добавит список файлов</p>



		<p>с расширением "doc" в файл "1.txt"</p> <p>b. Изменит расширения файлов с "doc" на "txt"</p> <p>c. Содержит ошибку</p> <p>d. Переименует файл "*.doc" в "1.txt"</p>
10.	Следующая команда в сценарии:attrib +R %1	<p>a. Добавит атрибут "Только чтение" для файла, имя которого передано в первом параметре</p> <p>b. Добавит атрибут "Только чтение" для всех файлов, имена которых содержат "1"</p> <p>c. Содержит ошибку</p> <p>d. Запросит у пользователя имя одного файла и атрибут для установки</p>
11.	Следующая команда в сценарии (.bat):echo %0	<p>a. Выведет на экран имя сценария</p> <p>b. Выведет на экран %0</p> <p>c. Выведет на экран 0</p> <p>d. Выведет на экран первый переданный сценарию параметр</p>
12.	Microsoft Windows использует:	<p>a. Страничное распределение памяти</p> <p>b. Сегментное распределение памяти</p> <p>c. Распределение памяти перемещаемыми разделами</p>
13.	Виртуальная память позволяет:	<p>a. Защитить процессы друг от друга</p> <p>b. Обеспечить предсказуемость времени доступа к памяти</p> <p>c. Обойтись без аппаратной поддержки при управлении памятью</p>
14.	Виртуальная страница при страничном распределении памяти:	<p>a. Располагается либо в физической памяти, либо во внешней памяти, либо отсутствует</p> <p>b. Может располагаться исключительно во внешней памяти</p> <p>c. Не может отсутствовать, все адреса виртуального адресного пространства всегда доступны</p>



15.	Для преобразования виртуального адреса в физический при страничном распределении нет необходимости учитывать:	a. Тип содержимого страницы, например исполняемый код или данные b. Размер страницы c. Данные из таблицы страниц процесса d. Смещение внутри виртуальной страницы
16.	Модули оперативной памяти на материнской плате, это:	a. Физическая память b. Виртуальная память c. Внешняя память d. Регистры процессора
17.	Недостатком виртуальной памяти является:	a. Непредсказуемое время доступа к памяти b. Процесс, требующий памяти больше, чем реально есть в системе загрузить нельзя c. Фрагментация памяти
18.	Недостатком оверлеев является:	a. Необходимость для прикладной программы самостоятельно реализовывать управление памятью b. Необходимость целиком загружать и выгружать образ процесса c. Процесс, требующий памяти больше, чем реально есть в системе, загрузить нельзя
19.	Почему размер страницы при страничном распределении выбирается кратным степени двойки?	a. Так из виртуального адреса проще выделить номер страницы и смещение b. Это просто традиция, которая на практике не дает никаких преимуществ c. Размер страницы выбирается по размеру кластера на диске
20.	Преимуществом виртуальной памяти является:	a. Возможность скрыть от прикладной программы реальную организацию памяти b. Простота реализации c. Нет необходимости аппаратной поддержки
21.	При использовании виртуальной памяти:	a. Программа может



		использовать памяти больше, чем объем физической (оперативной) памяти b. Доступны средства для создания виртуальной реальности c. Размер выделенной памяти фиксируется в момент запуска программы и в дальнейшем не может быть изменен
22.	При страничном распределении памяти:	a. Программист работает с одним линейным адресным пространством b. Исполняемый код и данные могут храниться в отдельных адресных пространствах c. Адресное пространство приложения не может превышать размеры физической памяти
23.	При страничном распределении памяти:	a. Память делится на страницы одинакового размера b. Память делится на страницы произвольного размера c. Размер любой страницы может измениться в процессе работы приложения
24.	В системе используется страничное распределение памяти. Процесс имеет 8-битное адресное пространство. Размер страницы - 4 байта. Сколько всего страниц у процесса?	a. 64 b. 16 c. 256 d. 4 e. 32
25.	В системе используется страничное распределение памяти. Процесс имеет 8-битное адресное пространство. Размер страницы - 8 байт. Сколько всего страниц у процесса?	a. 64 b. 16 c. 256 d. 4 e. 32
26.	В системе используется страничное распределение памяти. Процесс имеет 8-битное адресное пространство. Размер страницы - 32 байта. Сколько всего страниц у процесса?	a. 64 b. 16 c. 256 d. 8 e. 32
27.	Свопинг предполагает, что:	a. Образы процессов выгружаются на диск и



		возвращаются в оперативную память целиком b. Внешняя память не используется c. Процесс самостоятельно управляет распределением памяти
28.	Страничное распределение памяти предполагает:	a. Использование внешней памяти b. Возможность управлять физической памятью для каждого процесса c. Деление адресного пространства на части произвольного размера
29.	Алгоритмы планирования потоков, где решения о смене и выборе активного потока принимаются операционной системой называются:	a. Вытесняющими b. Невытесняющими c. Системными
30.	Алгоритмы планирования потоков, где потоку позволяет выполняться, пока он сам, по собственной инициативе не отдаст управление операционной системе, называются:	a. Вытесняющими b. Невытесняющими c. Системными
31.	Критическая секция это:	a. Участок исполняемого кода b. Данные c. Конструктивная часть процессора
32.	Недостатком систем разделения времени является:	a. Накладные расходы при переключении между задачами b. Отсутствие интерактивности c. Невозможность распараллелить вычисления и ввод-вывод
33.	Особенностью систем реального времени является:	a. При проектировании закладывается запас вычислительной мощности b. Предназначены для решения вычислительных задач c. Основной критерий при проектировании - удобство пользователя
34.	Преимуществом систем пакетной обработки является:	a. Хорошая пропускная способность (которая выражается в количестве задач, выполняемых за



		единицу времени) b. Интерактивность c. Пригодность для систем реального времени
35.	Выберите правильное утверждение:	a. Несколько потоков могут принадлежать одному процессу b. В настоящее время процесс и поток - это одно и то же c. Процесс не может иметь более одного потока
36.	Возможность параллельного выполнения вычислений и ввода-вывода позволяет:	a. Уменьшить суммарное время выполнения нескольких задач b. Уменьшить время выполнения каждой задачи c. Использовать центральный процессор для прямого управления вводом-выводом
37.	В операционной системе Windows каталоги (например при использовании файловой системы FAT):	a. Образуют древовидную структуру b. Образуют сетевую структуру c. Изолированы, один каталог не может находиться внутри другого
38.	Особенностью использования файловых систем в ОС Windows является:	a. Файловые системы используются независимо (автономно) b. Файловые системы монтируются в единое дерево c. Может использоваться ровно одна файловая система
39.	В файловой системе FAT в записи каталога о файле хранится:	a. Номер первого кластера b. Список всех кластеров c. Смещение начала файла в байтах относительно начала области данных
40.	Если кластер в файловой системе FAT не последний в файле, в соответствующем ему индексном указателе хранится:	a. Номер следующего кластера b. Номер предыдущего кластера c. Специальное значение "кластер занят"
41.	Журналирование файловой системы позволяет:	a. Сохранить целостность



		файловой системы b. Узнать, кто и когда создавал, удалял и редактировал файлы c. Сохранить данные, находящиеся в кэше при отключении питания
42.	Монтирование предполагает:	а. Объединение нескольких файловых систем в единое дерево b. Удаление всех файлов и создание новой файловой системы c. Создание специальных файлов, ассоциированных с устройствами ввода-вывода
43.	Особенностью файловой системы FAT является:	а. Корневой каталог занимает фиксированную область b. Файл всегда располагается в смежных кластерах c. Размер кластера не фиксирован и выбирается динамически, в процессе записи файла на диск.
44.	Особенностью файловой системы FAT является:	а. Файл всегда занимает целое число кластеров, поэтому дисковое пространство обычно используется неоптимально b. В одном кластере может храниться два и более файлов c. Кластеры используются только для хранения каталогов
45.	Особенностью жесткой ссылки является:	а. Файл не будет удален, пока имеется хотя бы одна жесткая ссылка b. Различают первичную жесткую ссылку и второстепенные c. Файл полностью удаляется, если удалить любую жесткую ссылку
46.	Особенностью ярлыка Windows является:	а. С точки зрения файловой системы ярлык является обычным файлом b. При удалении ярлыка



		файл также будет удален с. Файл не будет удален, пока есть хотя бы один ярлык, указывающий на него
47.	Отличие FAT16 от FAT32 состоит в:	a. Разрядности индексных указателей b. Разрядности операционной системы, под которой используется данная файловая система с. Размёре кластера: 16 или 32 байта
48.	Для получения списка кластеров файла в FAT необходимо:	a. Восстановить цепочку кластеров с помощью таблицы индексных указателей (FAT-таблицы) b. Восстановить цепочку кластеров, прочитав по очереди их содержимое с. Прочитать запись в каталоге для данного файла
49.	При появлении испорченной области на жестком диске, использующем файловую систему FAT:	a. Соответствующие кластеры помечаются в таблице индексных указателей специальным значением b. В особую область добавляется список плохих кластеров с. Невозможно далее использовать диск
50.	Примером корневого каталога в Windows в файловой системе FAT является:	a. C:\ b. C:\WINDOWS с. C:\SYSTEM
51.	Фрагментация в файловой системе может означать, что:	a. Соседние участки файла располагаются в несмежных участках диска b. Файл хранится в двух и более смежных кластерах с. Файл занимает меньше одного кластера
52.	Что такое каталоги?	a. это системные файлы, обеспечивающие поддержку структуры файловой системы b. это фиктивные файлы, ассоциированные с устройствами ввода-вывода с. циклические буферы,



		позволяющие выходной файл одной программы соединить со входным файлом другой программы d. это обычные файлы, отображенные на адресное пространство процесса по указанному виртуальному адресу
53.	В JScript сценарии имеется конструкция: <code>for(i in B) S+=B[i];B</code> в данном случае:	a. К переменной S будет добавлена сумма всех элементов массива B b. К переменной S будет добавлено значение первого элемента массива B c. Используется неверный синтаксис для цикла for
54.	Главным объектом WSH, позволяющим создавать другие объекты, является:	a. WScript b. WMain c. Функция main()
55.	Для вывода значения переменной K в JScript можно использовать конструкцию:	a. WScript.Echo(K); b. Echo K; c. printf("%s",K);
56.	Для запуска приложения из WSH сценария можно использовать объект:	a. WScript.Shell b. Scripting.FileSystemObject c. WScript.StdOut
57.	Для получения имени пользователя в WSH сценарии можно использовать объект:	a. Wscript.Network b. Scripting.FileSystemObject c. WScript.StdOut
58.	Для создания объекта в WSH используется функция:	a. WScript.CreateObject b. WScript.Echo c. Main.Create
59.	Для запуска WSH сценария C:\test.js в консольном режиме, в командной строке необходимо набрать:	a. cscript C:\test.js b. wscript C:\test.js c. C:\test.js
60.	Особенностью переменных JScript является:	a. Тип присваивается при записи значения b. Необходимо указать тип при объявлении c. В переменных нельзя хранить текст
61.	Преимуществом WSH по сравнению с пакетными файлами Windows является:	a. Наличие полноценных циклов и условий b. Возможность выполнить несколько консольных команд из одного сценария c. Возможность



		редактировать сценарий в текстовом редакторе
62.	Для того, чтобы в WSH запросить у пользователя данные в консоли, можно использовать функцию:	a. WScript.StdIn.ReadLine b. scanf c. Main.ReadFromConsole
63.	Выполнение какого требования к операционной системе позволяет пользователю выполнить свои приложения в новой операционной системе?	a. совместимости b. безопасности c. предсказуемости d. масштабируемости
64.	Микроядерная архитектура операционной системы предполагает, что:	a. В режиме ядра работает минимально необходимая часть операционной системы b. Возможности операционной системы сильно ограничены c. Отсутствует аппаратная поддержка операционной системы
65.	Операционные системы с многослойной архитектурой, по сравнению с монолитной, имеют следующее преимущество:	a. Проще отлаживать, дорабатывать, модифицировать b. Имеют большую производительность c. Проще проектировать
66.	Операционная оболочка (в отличие от операционной среды и операционной системы) определяет:	a. Интерфейс пользователя b. Особенности управления ресурсами системы c. Интерфейс прикладного программирования
67.	Операционная система (в отличие от операционной среды и операционной оболочки) определяет:	a. Особенности управления ресурсами системы b. Интерфейс пользователя c. Интерфейс прикладного программирования
68.	Операционная среда определяет:	a. Интерфейс прикладного программирования b. Интерфейс пользователя c. Особенности управления ресурсами системы
69.	Операционная система выполняет следующую функцию:	a. Все перечисленное b. Управление памятью c. Управление процессами d. Управление внешними устройствами



База тестовых вопросов для 4 семестра

№ п/п	Формулировка вопроса	Варианты ответов (полужирным шрифтом – верные варианты)
70.	При использовании флага CREATE_SUSPENDED в функции CreateThread:	a. Новый поток не может начать работу сразу после создания b. Запускается новый поток, текущий поток приостанавливается c. Новый поток не может быть приостановлен
71.	Параметр lpStartAddress функции CreateThread определяет:	a. Адрес функции, которая "реализует" поток b. Адрес возврата, на который должно передаться управление после завершения потока c. Идентификатор процесса, для которого создается поток
72.	Глобальные переменные:	a. Могут использоваться любым потоком процесса b. Могут использоваться только главным потоком c. Не могут использоваться потоками
73.	Каждый поток в Windows:	a. Имеет персональный стек b. Имеет личное пространство адресов виртуальной памяти c. На все время работы занимает в эксклюзивное пользование одно ядро процессора
74.	С потоком ассоциируется:	a. Контекст выполнения b. Виртуальное адресное пространство c. Файловые дескрипторы
75.	С процессом (в отличие от потока) ассоциируется:	a. Память b. Стек c. Процессорное время
76.	В вашей программе есть необходимость обеспечить синхронизацию потоков при доступе к пулу из 6 объектов. В этом случае целесообразно использовать:	a. Событие (Event) b. Семафор c. Мьютекс (Mutex)
77.	В вашей программе есть необходимость приостановить большое количество потоков до реакции пользователя. Т.е. после реакции пользователя сразу все потоки должны продолжить работу. В этом случае целесообразно использовать:	a. Событие (Event) b. Семафор c. Мьютекс (Mutex)
78.	Большинство современных операционных систем:	a. Поддерживают многопоточность b. Имеют невытесняющие алгоритмы многозадачности c. Не являются многозадачными



79.	Вам необходимо создать новый файл в ОС Windows. Однако, если файл уже существует, никакие действия не требуются, модифицировать существующий файл не нужно. В этом случае, в параметр <code>dwCreationDisposition</code> функции <code>CreateFile</code> необходимо передать:	a. CREATE_NEW b. CREATE_ALWAYS c. TRUNCATE_EXISTING
80.	Вы открываете файл в ОС Windows с помощью функции <code>CreateFile</code> . Для того, чтобы получить права на чтение и запись, необходимо параметр <code>dwDesiredAccess</code> заполнить как:	a. GENERIC_READ GENERIC_WRITE b. GENERIC_READ GENERIC_WRITE c. GENERIC_READ && GENERIC_WRITE d. GENERIC_READ & GENERIC_WRITE e. GENERIC_WRITE
81.	Параметр <code>dwShareMode</code> функции <code>CreateFile</code> в ОС Windows определяет:	a. Способ совместного доступа с другими процессами b. Поведение при открытии файла (например нужно ли создавать файл, если он не существует) c. Каким образом файл будет доступен по сети
82.	Фрагмент программы для ОС Windows <pre>if(FILE_ATTRIBUTE_READONLY==GetFileAttributes(L"file.dat")){ printf("OK");}</pre> проверит, что:	a. Файл имеет атрибут "только для чтения", и кроме него не имеет абсолютно никаких атрибутов b. Файл имеет атрибут "только для чтения" c. Файл НЕ имеет атрибута "только для чтения"
83.	Для обеспечения совместного доступа к открываемому файлу в ОС Windows (т.е. так, чтобы другие процессы также могли одновременно с нами получить к нему доступ) необходимо в параметре <code>dwShareMode</code> функции <code>CreateFile</code> передать:	a. 0 b. FILE_SHARE_READ FILE_SHARE_WRITE c. FILE_SHARE_READ & FILE_SHARE_WRITE
84.	В случае, если функция <code>CreateFile</code> в ОС Windows не смогла открыть файл, например по причине конфликта совместного доступа, возвращаемое значение будет равно:	a. INVALID_HANDLE_VALUE b. 0 c. Случайному значению
85.	Фрагмент программы для ОС Windows: <pre>int k=18; WriteFile(hFile,&k,sizeof(k),&Bytes,NULL);</pre> запишет в файл:	a. 4 байта, так как это размер типа int b. 2 байта, так как для записи числа 18 необходимо 2 символа c. 1 Кб, так как <code>WriteFile</code> не может манипулировать меньшими блоками
86.	Следующий фрагмент программы: <pre>int k = 186500; FILE *f = fopen("file1.txt", "w+"); fprintf(f, "%d", k);</pre>	a. 6 байт b. 2 байта c. 4 байта d. 8 байт



	запишет в файл:	
87.	Следующий фрагмент программы: <pre>int k = 23; FILE *f = fopen("file1.txt", "w+"); fprintf(f, "%d", k);</pre> запишет в файл:	a. 6 байт b. 2 байта c. 4 байта d. 8 байт
88.	При отсутствии флага LOCKFILE_FAIL_IMMEDIATELY, функция LockFileEx в ОС Windows:	a. Ждет, пока требуемый регион будет разблокирован b. Принудительно разблокирует требуемый регион файла c. Ничего не ждет, немедленно завершается в любом случае
89.	Функция CreateFile в ОС Windows позволяет:	a. Создать новый файл или открыть существующий b. Только создать новый файл c. Создать новый файл, но только текстовый
90.	Функция LockFileEx в ОС Windows:	a. Блокирует регион файла b. Шифрует файл c. Блокирует в любом случае весь файл целиком
91.	Функция SetFilePointer в ОС Windows используется для:	a. Установки позиции (номер байта), откуда данные файла будут считываться или записываться b. Создания ярлыка файла c. Отображения файла на память
92.	Функция WriteFile в ОС Windows позволяет:	a. Записать в файл произвольные данные, которые хранятся в памяти b. Записывать в файл только переменные, имеющие встроенные типы. Запись, например, структур невозможна c. Записывать в файл только те данные, которые ранее были считаны с помощью ReadFile
93.	Для эксклюзивного открытия файла в ОС Windows (т.е. так, чтобы другие процессы не могли одновременно с нами получить к нему никакого доступа) необходимо в параметре dwShareMode функции CreateFile передать:	a. 0 b. FILE_SHARE_READ FILE_SHARE_WRITE c. FILE_SHARE_READ & FILE_SHARE_WRITE
94.	Для чтения данных из отображенного на память файла в ОС Windows необходимо использовать:	a. Указатель, возвращенный MapViewOfFile b. Хэндл, возвращенный CreateFileMapping c. Функцию ReadFile совместно с адресом области отображения

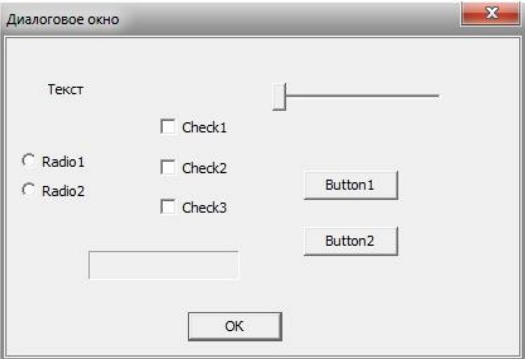


95.	Какая функция используется в процессе отображения файла на память в ОС Windows:	a. CreateFileMapping b. fopen c. WriteFile
96.	Какая функция (или функции) должна(ы) обязательно присутствовать в программе для отображения файла на память в ОС Windows:	a. Все перечисленные b. CreateFile c. CreateFileMapping d. MapViewOfFile
97.	Особенностью Microsoft Windows является:	a. Значительная часть адресного пространства процесса зарезервирована для использования системой b. Процессы используют 8-битные виртуальные адреса c. Каждый процесс занимает 2 Гб физической памяти вплоть до его завершения
98.	Отображение файла на память предполагает:	a. Можно работать с файлом как с участком памяти b. Файл полностью загружается в системный кэш c. Необходимо использовать SetFilePointer для доступа к файлу со сложной структурой
99.	Почему размер страницы при страничном распределении выбирается кратным степени двойки?	a. Так из виртуального адреса проще выделить номер страницы и смещение b. Это просто традиция, которая на практике не дает никаких преимуществ c. Размер страницы выбирается по размеру кластера на диске
100.	В системе используется страничное распределение памяти. Процесс имеет 8-битное адресное пространство. Размер страницы - 4 байта. Сколько всего страниц у процесса?	a. 64 b. 16 c. 256 d. 4 e. 32
101.	В системе используется страничное распределение памяти. Процесс имеет 8-битное адресное пространство. Размер страницы - 8 байт. Сколько всего страниц у процесса?	a. 64 b. 16 c. 256 d. 4 e. 32
102.	В системе используется страничное распределение памяти. Процесс имеет 8-битное адресное пространство. Размер страницы - 32 байта. Сколько всего страниц у процесса?	a. 64 b. 16 c. 256 d. 8 e. 32
103.	Файл состоит из записей (структур) фиксированного размера,	a. Массив из соответствующих



	которые записаны строго одна за другой. В этом случае после отображения файла на память он будет выглядеть как:	структур b. Двусвязный список из соответствующих структур c. Стек из соответствующих структур
104.	Функция GetSystemInfo в ОС Windows позволяет:	a. Получить общую информацию о памяти и процессоре b. Получить список потоков c. Получить имя компьютера, домена и текущего пользователя
105.	Функция VirtualLock в ОС Windows позволяет:	a. "Закрепить" страницы в физической памяти b. Сделать регион виртуальной памяти временно недоступным c. Отключить механизм виртуальной памяти
106.	WinAPI функция CreateWindow:	a. Может создать радиокнопку b. Регистрирует оконный класс и создает окно c. Имеет 4 параметра и может создать модальное окно с кнопкой "OK"
107.	В Windows окнами не являются:	a. Семафоры b. Кнопки c. Выпадающие списки d. Элементы панели задач
108.	Вы закрываете приложение Microsoft Word. Вместо того, чтобы закрыться, приложение в появившемся окне спрашивает, нужно ли сохранить изменения. Все остальные функции Word недоступны, до тех пор, пока Вы не ответите на вопрос. Данное окно (где спрашивается о сохранении изменений) является:	a. Модальным b. Элементом управления c. Новым приложением, по сути это не совсем окно
109.	Выберите правильный вариант:	a. Обычно приложение создает несколько окон b. Приложение может создать только одно окно. Для создания второго необходимо уничтожить первое. c. Окна не имеют иерархии, система поддерживает "плоский" список всех окон
110.	Для каждого потока, создавшего хотя бы одно окно, создается очередь оконных сообщений. Для того, чтобы извлечь сообщение из очереди, используется:	a. Функция GetMessage b. Программное прерывание c. Аппаратное прерывание d. Базовая система ввода-вывода (BIOS)
111.	Для создания оконного класса (в WinAPI):	a. Необходимо зарегистрировать его с помощью "RegisterClassEx" b. Необходимо расширить класс Window, например "class MyWindow: public Window"



		<p>с. Необходимо создать любое окно, его имя и будет именем класса</p>
112.	 <p>В данном случае окном не является:</p>	<p>а. Кнопка "закрыть" (красный крестик) б. Надпись "Текст" с. Кнопка "OK" д. Radio1</p>
113.	<p>Оконное сообщение отправляется:</p>	<p>а. Поток, который создал окно б. Процессу, который создал окно с. Первому свободному потоку д. Системному потоку, который обрабатывает все сообщения в своем контексте</p>
114.	<p>Оконный класс:</p>	<p>а. Определяет обработчик сообщений окна б. Не может быть одинаковым для двух окон с. Не может создаваться пользовательскими приложениями</p>
115.	<p>Особенностью модального окна является:</p>	<p>а. Невозможно работать с родительскими окнами, пока модальное окно не будет закрыто б. Модальные окна не могут быть созданы пользовательскими приложениями с. Для создания модального окна необходимо обязательно использовать ресурсы в исполняемом модуле</p>
116.	<p>Особенностью окон - элементов управления является:</p>	<p>а. Не могут работать без окна-родителя б. Не обрабатывают оконные сообщения с. Не могут сами обрабатывать сообщения мыши</p>
117.	<p>Пользователь нажал, а затем отпустил левую кнопку мыши, в то время, пока курсор находился над кнопкой:</p>	<p>а. Кнопка является окном, поэтому соответствующему обработчику будут отправлены сообщения б. Ядро Windows выполнит все необходимые действия. Приложение никогда не узнает о том, что была нажата кнопка</p>



		с. Будет вызвано программное прерывание, ранее зарегистрированное приложением
118.	После создания окон, поток должен:	a. Запустить цикл обработки сообщений b. Немедленно завершиться c. Вызвать функцию Sleep, так как после завершения потока все окна будут уничтожены
119.	Поток создал окна и был приостановлен (ждет завершения ввода-вывода):	a. Окна не будут функционировать b. Оконные сообщения будут переданы главному потоку c. При появлении в очереди оконного сообщения поток будет принудительно возобновлен, а операция ввода-вывода отменена
120.	Поток создал окна и завершился:	a. Окна потока будут уничтожены b. Окна потока будут переданы главному потоку c. Окна будут автоматически уничтожены только после завершения процесса d. Окна никогда автоматически не удаляются
121.	Предопределенные оконные классы:	a. Позволяют унифицировать пользовательский интерфейс b. Увеличивают размер исполняемых модулей пользовательских программ c. Ускоряют работу системы, так как оконные сообщения аппаратно обрабатываются видеоконтроллером
122.	Смысл параметра оконного сообщения Win32:	a. Различается в каждом конкретном случае b. Идентификатор элемента управления c. Идентификатор события элемента управления
123.	Сообщение WM_CREATE отправляется:	a. Только что созданному окну b. Окну родителю от только что созданного окна-потомка c. Любому потоку один раз при его создании
124.	Сообщение может быть отправлено окну в случае:	a. Во всех перечисленных b. Пользователь перемещает указатель мыши над окном c. Пользователь изменяет размеры окна



		d. Окно только что было создано
125.	Функция DispatchMessage:	a. Передает оконное сообщение обработчику b. Уничтожает выбранное окно c. Отправляет сообщение окну (сообщение будет помещено в очередь для обработки) d. Удаляет ошибочно отправленное сообщение
126.	Раздел реестра HKEY_CURRENT_USER:	a. Различается для каждого пользователя b. Формируется во время установки системы, ни один его параметр не может быть в дальнейшем изменен c. Хранится в %SystemRoot%\system32\config\system
127.	HKEY_CURRENT_USER это:	a. Корневой раздел b. Всегда пустой раздел c. Раздел, недоступный прикладным программам
128.	Раздел реестра HKLM\SAM:	a. Хранит данные, используемые диспетчером учетных записей b. Хранится в %USERPROFILE%\ntuser.dat c. Является ссылкой на HKLM\SOFTWARE
129.	В каждом разделе (key) реестра Windows:	a. Может быть задан безымянный параметр b. Может содержаться не более 10 параметров c. Может содержаться только один вложенный ключ
130.	В реестре Windows:	a. Параметры (values) вложены в разделы (keys) b. Разделы (keys) вложены в параметры (values) c. Разделы (keys) и параметры (values) - это одно и то же
131.	Для хранения параметров и настроек в Windows 7 в основном используется:	a. Реестр b. Централизованное хранилище *.ini файлов c. Набор *.ini файлов для каждого приложения d. Реляционная база данных на основе MSSQL
132.	Кусты реестра (hives):	a. Загружаются из файла или формируются динамически



		<p>b. Загружаются исключительно из файлов c. Всегда формируются динамически d. Использовались только в старых версиях Windows (3.1)</p>
133.	Реестр Windows представляет собой:	<p>a. Иерархическую базу данных b. Реляционную базу данных c. Неструктурированный набор нетипизированных данных d. Динамическую базу данных, хранимую исключительно в оперативной памяти, никакие компоненты которой нельзя сохранить на жесткий диск</p>
134.	Процесс SMSS:	<p>a. Запускается одним из первых процессов b. Создает окна проводника, кнопку "Пуск", часы на панели задач c. Запрашивает имя пользователя и пароль d. Отвечает за эффекты Windows Aero</p>
135.	System Idle Process (Бездействие системы):	<p>a. Может использовать технологию пониженного энергопотребления процессора b. Использовался в Windows 3.11, в настоящее время не используется c. Может быть закрыт с помощью диспетчера задач</p>
136.	В Windows под драйвером понимается:	<p>a. Как низкоуровневый драйвер аппаратуры, так и высокоуровневый драйвер, например файловой системы b. Только низкоуровневые драйверы, обрабатывающие аппаратные прерывания c. Только высокоуровневые драйверы, обрабатывающие программные прерывания</p>
137.	В качестве приложения-службы (service) Windows:	<p>a. Может использоваться приложение, имеющее специальные функции для обработки сообщений SCM b. Может использоваться любое приложение, например калькулятор c. Может использоваться только приложение, имеющее электронную цифровую подпись Microsoft</p>
138.	В случае, если в системе отсутствуют потоки, готовые к	<p>a. Управление передается потоку</p>



	выполнению:	"System Idle Process" b. Один из потоков будет принудительно возобновлен, даже если он ожидает завершения операции ввода-вывода c. Такая ситуация теоретически невозможна d. Ядро создаст новые потоки, готовые к выполнению
139.	Выберите приложение, которое вероятнее всего будет работать как служба (service) Windows:	a. DHCP сервер b. Блокнот c. Microsoft Word
140.	Выберите процесс, обеспечивающий работу с командной строкой:	a. CSRSS b. svchost.exe c. winlogon.exe d. dwm.exe
141.	Диспетчер управления службами представлен процессом:	a. services.exe b. winlogon.exe c. explorer.exe
142.	За авторизацию локальных пользователей отвечает процесс:	a. LSASS b. explorer.exe c. System Idle Process
143.	Настройки служб (services) Windows:	a. Хранятся в реестре b. Всегда хранятся в *.ini файлах c. Хранятся во внутренних структурах ядра и не могут быть изменены после установки операционной системы
144.	Особенностью драйверов Windows является:	a. Общая структура драйверов любого уровня b. Все драйверы должны иметь цифровую подпись Microsoft c. Любая программа может работать в качестве драйвера, даже, например, калькулятор (правда не имея при этом полезной функциональности)
145.	Особенностью модели ввода/вывода Windows является:	a. Запрос ввода/вывода последовательно обрабатывается стеком из нескольких драйверов. b. В системе фактически работает один драйвер, обеспечивающий всю функциональность ввода-вывода c. Система ввода-вывода может обойтись вообще без драйверов благодаря технологии Plug and Play
146.	Пакет IRP(I/O Request Packet):	a. Состоит из двух частей: "фиксированной" части и стека



		ввода/вывода. b. Имеет такой же формат, как у оконного сообщения c. Имеет текстовый формат, как например HTTP
147.	Почему в системе обычно работает несколько экземпляров процесса svchost.exe ?	a. Службам Windows требуются различные уровни доступа b. Каждому ядру процессора соответствует экземпляр svchost.exe c. В системе работает только один экземпляр svchost.exe, остальные представляют собой резерв на случай его внезапного завершения
148.	Службы Windows (service):	a. Запускаются независимо от пользователя, вошедшего в систему b. Запускаются только после входа пользователя в систему c. Использовались в Windows 3.11, в настоящий момент считаются устаревшими
149.	Службы Windows (service):	a. По умолчанию запускаются от имени пользователя LocalSystem b. Запускаются всегда от имени пользователя, вошедшего в систему c. Не ассоциируются ни с какой учетной записью, работают в контексте ядра
150.	Службы Windows (services) являются аналогом:	a. Демонов UNIX b. Зомби-процессов UNIX c. Поточков UNIX
151.	Типичная функциональность драйвера:	a. Обработка прерываний b. Создание окон c. Обеспечение пользовательского интерфейса
152.	Что произойдет, если завершить процесс explorer.exe?	a. Исчезнет панель задач b. Завершить его не получится c. Больше ни один пользователь не сможет войти в систему d. Произойдет перезагрузка компьютера
153.	UNICODE позволяет:	a. Кодировать символы для множества языков, в том числе кириллицу b. Реализовать WinAPI так, что приложение работает напрямую с ядром c. Преобразовывать имя файла в Хэндли



154.	WinAPI позволяет:	a. Унифицировать доступ приложений к различного рода ресурсам b. Ускорить работу системы за счет прямого доступа пользовательских программ к аппаратному обеспечению c. Контролировать все приложения, т.к. приложение не сможет запуститься, если не создаст хотя бы одно окно
155.	WinAPI представляет собой:	a. Набор функций для взаимодействия с ОС и выполнения типичных простых операций b. Аналог ядра UNIX, работающий в привилегированном режиме c. Набор вспомогательных функций, статически (без использования dll) линкуемых к исполняемому модулю, облегчающих программисту взаимодействие с Windows d. Устаревший механизм, позволяющий приложению напрямую обращаться к физической памяти
156.	WinAPI реализуется:	a. С помощью динамически подключаемых библиотек b. С помощью статически подключаемых библиотек (через компоновщик) c. С помощью аппаратных прерываний
157.	В какой версии Windows поддерживается разграничение доступа к файлам:	a. Windows 2000 b. Windows Millennium Edition c. Windows 98 d. Windows 3.11
158.	В результате ошибки процесс вместо хэндла использовал адрес в памяти, содержащий "мусор" (некоторое произвольное значение):	a. Процесс может обратиться к другому, ранее открытому им же ресурсу b. Может быть нарушена работа произвольного процесса, даже системного c. С большой вероятностью будет нарушена работа ядра, потребуются перезагрузка системы
159.	Для выполнения действий с файлом приложение использует:	a. Хэндл файла b. Общесистемный идентификатор файла



		с. Номер записи MFT
160.	Для идентификации открытого файла внутри ядра Windows используется:	a. Уникальный общесистемный идентификатор b. Хэндл c. Тот же идентификатор, что и у процесса, открывшего файл. Таким образом, все ресурсы одного процесса имеют одинаковые идентификаторы.
161.	К семейству Windows 9x относится:	a. Windows Millennium Edition b. Windows NT c. Windows Vista
162.	К семейству Windows NT не относится:	a. Все относятся b. Windows XP (64-bit) c. Windows XP d. Windows 2000
163.	Какая из перечисленных WinAPI функций работает с UNICODE:	a. CreateFileW b. CreateFileA c. Ни одна из перечисленных
164.	Под какой ОС работает приложение, если ему предоставляется 3 способа взаимодействия с аппаратурой: напрямую помощью BIOS, помощью операционной системы через известные программные прерывания	a. DOS b. Windows XP c. Windows 98 d. OS/2
165.	Подсистемы окружения Windows:	a. В настоящее время неактуальны, т.к. используется только одна подсистема b. Позволяют запускать под Windows программы, собранные под Linux или OS/2, без дополнительных изменений c. Позволяют ОС Windows и приложениям использовать новые возможности процессоров IA64 и AMD64
166.	Пользовательские приложения для Windows:	a. Работают в пользовательском режиме процессора b. Работают в режиме ядра для увеличения быстродействия c. Используют прерывания для управления физической памятью
167.	Последние версии Windows семейства 3.11 не поддерживали:	a. Вытесняющую многозадачность b. Файловые системы c. Работу с окнами
168.	Последние версии Windows семейства 9x не поддерживали:	a. Квоты b. Вытесняющую многозадачность c. Стек протоколов TCP/IP d. Работу с окнами



169.	Последние версии Windows семейства 3.11 не поддерживали:	а. Журналируемые файловые системы b. Работу с сетью c. Работу в режиме графического интерфейса
170.	Приложение, написанное для Windows 98:	а. Во многих случаях сможет работать под Windows 7, так как для взаимодействия с ОС используется унифицированный интерфейс b. Сможет работать под DOS c. Сможет работать под Windows 7, но только после перекомпиляции d. Может работать исключительно под Windows 98
171.	Процесс открыл файл и передал хэндл файла другому процессу:	а. Хэндл не будет работать в контексте другого процесса b. Оба процесса смогут совместно использовать файл c. Произойдет ошибка, оба процесса будут завершены, система предложит отправить сообщение в Microsoft
172.	К фиксированным "процессам поддержки системы" можно отнести:	а. Процесс "Winlogon", обеспечивающий вход пользователя в систему b. Процесс "calc.exe" (калькулятор), так как программа присутствует во всех версиях Windows c. Web-сервер в Windows 2008, так как он запускается после загрузки системы, даже если ни один пользователь не вошел в систему
173.	Функции WinAPI, использующие текстовые строки:	а. Имеют две реализации: UNICODE и ANSI b. Должны использовать идентификатор для работы со строкой, так как строка является объектом ядра c. Должны использовать хэндл окна, так как строка - это всегда окно
174.	Хэндл представляет собой:	а. Объект для доступа к ресурсам, истинная сущность которого слабо документирована b. Общесистемный идентификатор ресурса, который использует ядро для учета ресурсов c. Специальный файл, ассоциированный с ресурсом



175.	Адрес раздела PE файла в виртуальной памяти определяет:	a. Таблица объектов b. Необязательный заголовок c. Обязательный заголовок d. Заголовок DOS
176.	Заголовок DOS в PE файле:	a. Позволяет вывести предупреждение при попытке запустить Windows-приложение под DOS b. Использовался в самых ранних версиях PE, в настоящее время отсутствует c. Используется для загрузки Windows-приложений, так как DOS и Windows приложения мало чем отличаются
177.	Заголовок Windows PE файла содержит:	a. Обязательный и необязательный заголовки b. Сигнатуру и 7 разделов c. Заголовок DOS, заглушку и необязательный заголовок
178.	Какие файлы имеют формат PE:	a. .exe и .dll b. Только .exe c. Только .dll d. .doc, .ppt и .xls e. Все перечисленные
179.	Особенность загрузки PE модуля в память:	a. Модуль должен быть загружен по определенным адресам, но при необходимости может быть перемещен b. Модуль всегда загружается по заранее заданному адресу c. Приложение, использующее модуль, всегда самостоятельно выбирает адрес для загрузки дополнительного модуля
180.	Сигнатура заголовка DOS (DOS_magic) PE файла соответствует символам:	a. MZ b. PE c. XZ d. EXE
181.	Символы сигнатуры MZ расшифровываются как:	a. Марк Збиковски b. Мозамбик c. Microsoft Zapper
182.	Суть данных раздела PE файла, загруженного в виртуальную память определена:	a. В необязательном заголовке, в частности в массиве DataDirectory b. В таблице объектов c. В заголовке DOS d. В сигнатуре файла
183.	Таблица объектов PE файла определяет:	a. Место расположения раздела в



		файле b. Адрес структур экспорта в оперативной памяти c. Размеры заголовков файла
184.	Таблица объектов PE файла располагается:	a. Сразу после заголовка Windows b. По смещению DOS_ifanew c. В самом начале файла d. Внутри заголовка DOS
185.	При использовании LVM "Logical Volume" это:	a. Несколько физических томов b. Логический раздел, который можно отформатировать и использовать как обычный раздел жёсткого диска c. Область размером 4 МБ на физическом томе
186.	LVM позволяет:	a. Перемещать логические разделы с одного физического устройства на другое "на лету" b. Использовать коды Хэмминга для исправления ошибок c. Совмещать локальное и облачное хранилище
187.	LVM позволяет:	a. Создавать копии (снимки) логического тома на лету b. Использовать коды Хэмминга для исправления ошибок c. Совмещать локальное и облачное хранилище
188.	LVM это:	a. Дополнительный слой абстракции, позволяющий собрать несколько разнородных дисков в один, и затем снова разбить его так, как необходимо b. Виртуальный диск, который использует для хранения данных раздел жесткого диска и специально выделенные страницы оперативной памяти c. Подсистема, управляющая виртуальной памятью
189.	При использовании LVM "volume group" это:	a. Несколько физических томов b. Несколько логических томов c. Область на физическом томе
190.	Одним из принципов работы OOMKiller является:	a. Завершить минимальное количество процессов b. Завершить максимальное количество процессов c. В первую очередь завершаются процессы, потребляющие мало



		памяти
191.	В Linux параметр <code>vm.overcommit_memory</code> управляет:	a. "Агрессивностью" использования SWAP b. Размером SWAP-памяти c. Возможностью выделять памяти больше, чем реально есть в системе
192.	В Linux параметр <code>vm.swappiness</code> управляет:	a. "Агрессивностью" использования SWAP b. Размером SWAP-памяти c. Возможностью выделять памяти больше, чем реально есть в системе
193.	В Linux с ядром версии 3.10 можно использовать:	a. Раздел подкачки и файл подкачки b. Только раздел подкачки, но не файл подкачки c. Только файл подкачки, но не раздел подкачки
194.	Вы выделили память командой <code>malloc</code> , но еще не производили операций чтения или записи с этими адресами. Для ядра действуют настройки по умолчанию. Выберите правильный ответ:	a. Адреса зарезервированы в виртуальном адресном пространстве процесса. Страницы физической памяти и swap могут быть пока не задействованы b. Выделены страницы физической памяти в соответствии с запрашиваемым размером выделяемой памяти, однако swap может быть пока не задействован c. Память выделена в разделе подкачки, физическая память пока не используется
195.	<code>pdflash</code> выполняет следующую задачу:	a. Запись "грязных" данных на жесткий диск b. Перемещение данных между swap и физической памятью c. Поддержка Memory Overcommit
196.	Особенностью планировщика CFQ является:	a. Каждому процессу присваивается собственная очередь b. Используется простая очередь FIFO, запросы исполняются без переупорядочивания c. Используются три очереди: одна отсортированная и две неотсортированных
197.	Особенностью планировщика Deadline является:	a. Каждому процессу присваивается собственная очередь b. Используется простая очередь FIFO, запросы исполняются без



		переупорядочивания с. Используются три очереди: одна отсортированная и две неотсортированных
198.	Вы установили параметр <code>vm.dirty_background_ratio = 20</code> Это означает:	а. После заполнения 20% памяти "грязными" данными вывод будет приостановлен, начнется запись этих данных на жесткий диск б. После заполнения 20% памяти "грязными" данными начнется фоновая запись этих данных на жесткий диск с. "Грязные" данные могут находиться в кэше не более 20 секунд, после чего начнется фоновая запись этих данных на жесткий диск
199.	Вы установили параметр <code>vm.dirty_ratio = 70</code> Это означает:	а. После заполнения 70% памяти "грязными" данными вывод будет приостановлен, начнется запись этих данных на жесткий диск б. После заполнения 70% памяти "грязными" данными начнется фоновая запись этих данных на жесткий диск с. "Грязные" данные могут находиться в кэше не более 70 секунд, после чего начнется фоновая запись этих данных на жесткий диск
200.	Для добавления физического тома в группу томов LVM используется команда:	а. vgextend б. lvextend с. lvcreate
201.	Для инициализации диска или раздела для использования с LVM необходимо использовать команду:	а. pvcreate б. vgcreate с. lvcreate
202.	Для создания снимка логического раздела LVM можно использовать команду:	а. lvcreate б. vgchange с. pvscan
203.	Информация о RAID массивах в Linux находится в файле:	а. /proc/mdstat б. /proc/raid с. /proc/sys
204.	Команда <code>mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/hde1 /dev/hdf2</code>	а. Создаст RAID массив, у которого 50% дискового пространства будет использовано для обеспечения отказоустойчивости б. Создаст два RAID массива первого уровня с. Создаст RAID массив, где блоки



		четности равномерно распределяются по всем дискам
205.	Пометить диск в RAID массиве как сбойный можно командой:	a. mdadm b. fdisk c. mdstat
206.	Недостатком RAID1 является:	a. Потеря дискового объема b. Низкая скорость чтения c. Высокая вероятность потери данных
207.	Недостатком RAID0 является:	a. Высокая вероятность потери данных b. Низкая скорость записи c. Низкая скорость чтения
208.	Особенностью RAID5 является:	a. Контрольные суммы записываются на все диски массива b. Используются коды Хэмминга c. Данные разбиваются на байты, а не на блоки
209.	Отличие RAID4 от RAID3 состоит в следующем:	a. Данные расщепляются на блоки, а не на байты b. Для хранения блоков четности используется один диск, а не два c. Блоки четности хранятся на всех дисках, а не на одном
210.	Преимуществом RAID0 является:	a. Эффективное использование дискового пространства b. Высокая отказоустойчивость c. Блоки четности равномерно распределены по всем дискам
211.	Пакетный менеджер APT используется в операционной системе:	a. Ubuntu b. Gentoo c. SUSE Linux
212.	В CentOS используется менеджер пакетов:	a. yum b. yast c. apt
213.	Для поиска пакета в сетевом репозитории можно использовать команду:	a. yum search b. wget search c. make search
214.	Преимуществом сборки программного обеспечения из исходных кодов, по сравнению с установкой из rpm с помощью пакетного менеджера, является:	a. Возможность установить самую последнюю версию ПО b. Автоматическая установка всех зависимостей c. Поддержание базы данных всего ПО, которое установлено в вашей системе



215.	Выберите самый правильный способ установки rpm пакета, который расположен на жестком диске вашего компьютера:	a. yum localinstall b. rpm -i c. make install
216.	CentOS, это дистрибутив, основанный на:	a. Red Hat Enterprise Linux b. Windows 8.1 c. Mac OS
217.	Virtual Box позволяет:	a. Работать с виртуальной машиной, имитирующей реальный компьютер b. Создавать разделы на жестком диске, а также менять структуру таблицы размещения разделов c. Создавать загрузочный флеш-накопитель на основе дистрибутива в виде файла .iso
218.	Для входа в программу настройки BIOS необходимо:	a. Нажать клавишу или сочетание клавиш, предусмотренные производителем оборудования, на начальном экране загрузки b. Нажать кнопку "RESET" при выключенном компьютере, и не отпуская ее, включить компьютер c. Зайти в панель управления, выбрать пункт "Администрирование" и затем подпункт "Настройка BIOS"
219.	Для установки операционной системы на виртуальную машину в VirtualBox необходимо:	a. В настройках виртуальной машины указать путь к дистрибутиву в виде файла с расширением ".iso" b. Выбрать операционную систему и дистрибутив, дождаться, пока VirtualBox самостоятельно скачает нужный образ DVD диска c. Во время установки программы VirtualBox выбрать операционную систему. Эта операционная система начнет устанавливаться во время первого запуска.
220.	Какая утилита может использоваться для редактирования разделов жесткого диска?	a. GNOME Partition Editor b. UNetbootin c. Программа настройки BIOS
221.	MFT это:	a. Важный файл NTFS, в котором хранится информация о всех файлах раздела b. Формат таблицы разделов жесткого диска c. Файловая система, которая может использоваться вместе с Linux



222.	В Protective MBR записано, что:	a. Диск содержит один большой раздел b. Диск не содержит ни одного раздела c. Четвертый раздел является активным
223.	Protective MBR выполняет следующую функцию:	a. Защищает GPT от повреждения старыми утилитами b. Определяет, какой из разделов является "EFI System" c. Хранит резервную копию таблицы разделов GPT
224.	По умолчанию UEFI передает управление:	a. Загрузчику /EFI/Boot/bootx64.efi в разделе EFI System b. Загрузчику Protective MBR c. Загрузчику последнего раздела
225.	Для загрузки с диска, имеющего формат GPT, обязательно наличие следующей системы:	a. UEFI b. BIOS c. CHS
226.	Код загрузчика MBR занимает:	a. Первые 446 байт диска b. Первые 512 секторов диска c. Последний сектор первого раздела
227.	Максимальный размер файла для FAT32 :	a. составляет ~4Гб b. не ограничен c. составляет ~4Тб
228.	Механизм адресации для жесткого диска, использующий 48-битное число - порядковый номер блока, называется:	a. LBA b. CHS c. LARGE
229.	Недостаток формата таблицы разделов GPT состоит в следующем:	a. GPT не поддерживается старыми операционными системами b. GPT позволяет создавать максимум 4 основных раздела c. GPT не позволяет работать с дисками размером больше 2 Тб.
230.	Первые 512 байт диска, имеющего формат GPT, занимает:	a. Protective MBR b. UEFI firmware c. FAT
231.	Жесткий диск объемом 500 Гб с точки зрения операционной системы Windows имеет объем всего 465 Гб. Причина этого несоответствия следующая:	a. Производители жестких дисков считают, что килобайте - 1000 байт, мегабайте - 1000 килобайт и т.д. Операционная система считает, что в килобайте - 1024 байта, мегабайте - 1024 килобайт и т.д. b. Продавцы жестких дисков обманывают потребителей



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Операционные системы» по направлению подготовки 09.03.04 «Программная инженерия» направленности «Разработка программно-информационных систем» ФГБОУ ВО «ЧелГУ»

стр. 40

		с. Разница в объеме - это место, занятое для размещения MBR
232.	Формат таблицы разделов, позволяющий создавать не более 4 основных разделов, называется:	a. MBR b. GPT c. CHS



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

Семестр 3:

Экзамен проводится в виде тестирования. Студент должен ответить на вопросы закрытого типа, которые предполагают выбор вариантов ответа, а также на вопросы открытого типа, которые не предполагают вариантов ответа, правильный ответ требуется написать самостоятельно. Всего 20 тестовых вопросов. Продолжительность теста – 35 минут.

Семестр 4:

Экзамен проводится в виде тестирования. Студент должен ответить на вопросы закрытого типа, которые предполагают выбор вариантов ответа, а также на вопросы открытого типа, которые не предполагают вариантов ответа, правильный ответ требуется написать самостоятельно. Всего 20 тестовых вопросов. Продолжительность теста – 35 минут.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Тест формируется в системе электронного обучения MOODLE.

Максимальный балл за тест — 100 баллов.

Оценка	Отлично/ Зачтено	Хорошо/ зачтено	Удовлетворител ьно/зачтено	Неудовлетворительно / незачтено
Баллы	100-90 баллов	89-75 баллов	74-60 баллов	59-0 баллов
Уровень освоения проверяемых компетенций	высокий	средний	базовый	недостаточный

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

При подведении итогов учитываются результаты только промежуточной аттестации:

0-59 баллов – неудовлетворительно/незачтено;

60-74 баллов – удовлетворительно/зачтено;

75-89 баллов – хорошо/зачтено;

90-100 баллов – отлично/зачтено;

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке отлично:
 - предполагает формирование компетенций на высоком уровне;



- знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки отлично;
 - студент умеет применять на практике знания, полученные в рамках изучения дисциплины
 - формируются навыки использования теоретических и практических разделов дисциплины для решения задач профессиональной деятельности;
2. Средний уровень соответствует оценке хорошо:
- предполагает формирование компетенций на среднем уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки хорошо;
 - студент умеет применять знания, полученные в рамках изучения дисциплины, для решения задач профессиональной деятельности;
3. Базовый уровень соответствует оценке удовлетворительно:
- предполагает формирование компетенций на базовом уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки удовлетворительно;
4. Недостаточный уровень соответствует оценке неудовлетворительно.