Министерство науки и высшего образования Российской Федерации Министерство науки, высшего образования и инноваций Кыргызской Республики

Межгосударственная образовательная организация высшего образования Кыргызско-Российский Славянский университет имени первого Президента Российской Федерации Б.Н. Ельцина.
Политехнический институт
Высшая школа цифровых технологий

ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине

СТРУКТУРА ОПЕРАЦИОННЫХ СИСТЕМ

Уровень высшего образования БАКАЛАВРИАТ

Направление подготовки
01.03.02 - РФ, 510200 - КР, Прикладная математика и информатика
(код и наименование направления подготовки)

Квалификация
Бакалавр

Бишкек 2026

Фонд оценочных средств предназначен для контроля знаний обучающихся по направлению подготовки (специальности) «Прикладная математика и информатика» по дисциплине

«СТРУКТУРА ОПЕРАЦИОННЫХ СИСТЕМ»

наименование

Фонд оценочных средств рассмотрен и утвержден на заседании высшей школы

Цифровые технологии

наименование высшей школы

протокол №1 от «02» 09. 2026г.

Директор высшей школы Цифровых технологий

А Хмелева Ирина Владимировна

наименование кафедры *подпись* *расшифровка подписи*

Руководитель образовательной программы

Доулбекова Салтанат Байызбековна Доцент

ФИО *должность* подпись

Исполнители:

Доцент Аширбаев Бейшембек Ыбышевич

расшифровка подписи подпись

Профессор Керимбеков Акылбек Керимбекович

расшифровка подписи подпись

Раздел 1. Перечень компетенций, с указанием этапов их формирования в процессе освоения образовательной программы

Формируемые компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций	Виды оценочных средств/ шифр раздела в данном документе
ПК-1. Способность осуществлять педагогическую деятельность по реализации программ углубленного изучения естественных наук среднего общего образования с учетом современных методик	**Знать:** - Основы общетеоретических дисциплин (педагогика, психология) в объеме, необходимых для решения педагогических, научно-методических и организационно-управленческих задач; - методику учебной и воспитательной работы; современных педагогических технологий реализации компетентностного подхода с учетом индивидуальных особенностей обучающихся; - методы и технологии развивающего обучения.	Блок А: СРС – лабораторная работа
	Уметь: - применять методы современных, информационных и цифровых образовательных технологий; - проводить учебные занятия опираясь на инновационные методы в области педагогических и психологических наук - использовать разнообразные формы и приемы в оценке образовательной и научной деятельности обучающегося; - организовать исследовательскую деятельность обучающихся, включая интерактивные методы.	Блок Б: тесты
	Владеть: - навыками формирования общекультурных компетенций в процессе обучения; - методами планирования специализированного образовательного	

	процесса для углубленного изучения предмета; - умениями формировании индивидуальных траекторий обучения для развития - обучающихся. предметным языком математики и навыками грамотного описания решения задач и представления полученных результатов; -навыками самостоятельной работы и освоение новых дисциплин;	

Раздел 2. Технологическая карта дисциплины

Технологическая карта дисциплины

«Структура операционных систем»

Курс/семестр: 4/8

Количество кредитов(ЗЕ): 3

Отчетность: Зачетно-экзаменационная ведомость (экзамен)

Название модулей дисциплины согласно РПД	Контроль	Форма контроля	зачетный минимум	зачетный максимум	график контроля
Модуль 1					
Архитектура операционных систем	Текущий контроль	Групповая дискуссия, тестирование	5	10	30
	Рубежный контроль	Практическое занятие, индивидуальные задания	5	10	
Модуль 2					
Файлы и файловая система ОС Unix.	Текущий контроль	Групповая дискуссия, тестирование	5	10	35
	Рубежный контроль	Практическое занятие, индивидуальные задания	5	10	
Модуль 3					
Процессы в ОС Unix.	Текущий контроль	Групповая дискуссия, тестирование	5	10	40
	Рубежный контроль	Практическое занятие, индивидуальные задания	5	10	
Модуль 4					
Блок управления процессами	Текущий контроль	Групповая дискуссия, тестирование	5	5	45
	Рубежный контроль	Практическое занятие, индивидуальные задания	5	5	

ВСЕГО за семестр		40	70	
Промежуточный контроль (Экзамен)	Тест.	20	30	
Семестровый рейтинг по дисциплине		60	100	

Модуль	логически завершенная часть дисциплины
Текущий контроль	самостоятельная работа студента, посещаемость и активность на занятиях
Рубежный контроль	проверка полноты знаний и умений по материалу модуля в целом
Промежуточный контроль	завершенная задокументированная часть учебной дисциплины - совокупность тесно связанных между собой модулей дисциплины.

Раздел 3. Типовые контрольные задания и иные материалы, необходимые для оценки планируемых результатов обучения по дисциплине. Описание показателей и критериев оценивания компетенций, описание шкал оценивания. Блок А.

СРС - Лабораторная работа "Работа с файлами"

Изучение файловой системы

Войдите в систему в режиме командной строки, используя данные имя и пароль. Система должна вывести приглашение командной строки (значок доллара, процента или тому подобное).

1.1 Получение справки

%**man** ***файл***

где ***файл*** - команда, программа, конфигурационный файл или библиотечная функция, по которой требуется получить справку. Справочник *man* состоит из нескольких разделов:

1: Команды пользователя

1M: Команды для системного администрирования

2: Системные вызовы

3 (3B,3S,...): Библиотечные функции

4: Форматы файлов

5: Заголовки, таблицы, макросы, разное

6: Демонстрации

7: Интерфейсы устройств и сети

При упоминании какой-либо команды, функции или системного файла в литературе по Unix часто в скобках указывается номер раздела справочника *man*, например: "команда *ls(1)* предназначена для вывода списка файлов". Как правило *man* сам находит раздел с требуемой справкой, просматривая все разделы по очереди, поэтому для получения справки по команде *ls* достаточно ввести

%**man ls**

Управление скролингом текста:

- <*пробел*> - страница вниз

- *<Enter>* - строка вниз
- *<Ctrl>+B* - страница вверх
- /**word** - поиск слова **word**
- *q* - выход

Однако иногда требуемый термин может находиться сразу в нескольких разделах (*printf* - команда и библиотечная функция); в этом случае *man* выдаст документ из первого найденного раздела. Для получения документа из другого раздела используется ключ -s *раздел*. Для получения списка разделов, в которых встречается требуемый термин используется ключ -l. Ключ -a дает указание вывести все найденные справки из всех разделов друг за другом.

Обращайте внимание на номера разделов (они указываются в заголовке каждой страницы).

%**man printf**

%**man -s 3S printf**

%**man -l printf**

%**man -a printf**

1.2 Изучение файловой системы

%**pwd** выводит полное имя текущего каталога (в котором вы находитесь).

%**ls** выводит список файлов текущего каталога.

%**ls –l** выводит список файлов с их атрибутами, например:

-rwxr-xr-- 1 vm97 students 24380 Feb 3 2000 hello

ABBBBBBBBB C DDDD EEEEEEEE FFFFF GGGGGGGGGGG HHHHH –

см. ниже описание полей листинга

A. Тип файла

-

обычный файл;

d

каталог (фактически, каталог - это специальный файл, в котором перечислены файлы этого каталога);

l

символическая связь (файл, содержащий имя другого файла - является сылкой на этот файл; подробности ниже в команде *ln*);

c

символьное устройство (устройство, с которым операции чтения-записи производятся побайтно);

b

блочное устройство (операции чтения-записи буферизуются). Некоторые устройства являются только символьными (например, последовательный порт), а некоторые представлены в обоих вариантах (например, диск).

p

конвейер (pipe; программный канал, организованный в виде очереди, для передачи данных между двумя процессами).

B. Права на использование файла

Следующие после типа файла 9 символов представляют собой права на использование файла: первая тройка символов - для пользователя-владельца файла, вторая - для группы-владельца, третья - для всех остальных. Наличие буквы в каждой тройке означает наличие соответствующего права:
для обычных файлов:

r - право на чтение содержимого файла

w - право на изменение содержимого файла

x - право на исполнение (запуск) файла

для каталогов:

r - право на получение списка файлов каталога

w - право на создание и удаление файлов в каталоге

x - право на "поиск" в каталоге

Набор прав --x для каталога позволяет войти в него и прочитать файл, имя которого заранее известно (при условии наличия прав чтения этого файла). Можно также выполнить команды *ls* и *ls -l* для этого файла. При этом получить список файлов (*ls (-l)* для каталога), а также создать или удалить файл нельзя.
Право r-- для каталога позволяет произвести чтение каталога, то есть получить список имен его файлов. При этом ни войти в каталог, ни узнать атрибуты файлов (*ls -l*), ни прочесть файл (даже при наличии прав чтения этого файла) нельзя. Создать или удалить файл, естественно, нельзя.
Что означают права -wx для каталога? Где это может применяться?
Модификация файла требует прав на запись в файл, так как это операция над файлом. Удаление файла требует прав на запись в каталог, в котором находится этот файл - это операция над каталогом. Наличие права записи в файл не обязательно означает возможность его удалить (хотя можно размер файла уменьшить до нуля).

C. Число связей

Число жестких связей (имён) файла - подробнее см. ниже команду *ln*. Для каталога - число его подкаталогов (как минимум два - '**.**' *точка* и '**..**' *две точки*).

D. Пользователь-владелец файла

E. Группа-владелец файла

Группа - это именованное множество пользователей; группы составляются администратором. Каждый пользователь является членом как минимум одной группы.
Пользователь-владелец файла (далее - просто владелец) не обязательно является членом группы-владельца, хотя и может быть таковым.

F. Размер файла в байтах

G. Дата последней модификации файла

H. Имя файла

%**ls -a**

ls -a выводит также файлы, имена которых начинаются с точки. По умолчанию эти файлы считаются "неинтересными" и не показываются. Во всех остальных аспектах это совершенно обычные файлы (кроме, конечно, файла '.' (точка), обозначающего текущий каталог, и файла '..' (две точки) обозначающего вышележащий каталог).

%**ls -al**

(обратите внимание на наличие символической связи)

Шелл bash для выполнения начальных установок пользователя в некоторых случаях читает файл **.profile**, а в некоторых - **.bashrc**. Создание символической связи повзоляет не создавать два одинаковых файла.

%**ls b.txt**

%**ls -l b.txt**

%**ls prog**

%**ls -l prog**

Проблема: как вывести атрибуты самого каталога, а не его содержимого. Возможные решения: *ls -l* (строка файла **prog**) и *ls -al prog* (строка файла '**.**') неудовлетворительны при большом числе файлов в текущем каталоге и в каталоге **prog** - получится слишком длинный листинг. Решение:

%**ls -d prog**

%**ls -ld prog**

%**ls -ld .**

%**ls -l ..**

%**ls -ld ..**

%**cd** ***каталог***

- переход в указанный каталог. Без указания каталога - переход в домашний.

1.3 Изучение содержимого файла

%**file** ***имя_файла*** **[*имя_файла* ...]**

определяет тип файла путем анализа его содержимого

Определите тип всех файлов вашего домашнего каталога и каталога **prog**.

Расширение файла (часть имени после точки) в Unix не имеет никакого специального значения. Например, для запуска файла требуется не расширение типа .exe, а право на исполнение.

%**cat** ***имя_файла***

- выводит содержимое файла на терминал.

%**cat b.txt**

%**more** ***имя_файла***

- пейджер (выводит файл постранично, с остановом). Фактически, команда *man* формирует текст справки и пропускает его через *more*. Поэтому команды управления прокруткой текста см. выше в описании *man*.

%**more b.txt**

%**tail [-*число*] *имя_файла***

выводит последние 10 строк файла или сколько указано.

%**tail b.txt**

%**tail -2 b.txt**

%**head [-*число*] *имя_файла***

выводит первые 10 строк файла или сколько указано.

%**head b.txt**

%**head -2 b.txt**

%**od** ***имя_файла***

выводит файл в восьмеричном виде. Ключи:

-N*число* - вывести только указанное число первых байт,

-j*число* - пропустить указанное число первых байт,

-j0x*число* - пропустить указанное число первых байт (число является 16-ричным),

-tx - вывести в шестнадцатеричном виде,

-c - вывести каждый байт в виде символа (непечатные символы отображаются как спецсимволы языка Си: \t,\n,\b,\f,\r,\0, либо как трехзначное 8-ричное число).

1.4 Манипулирование файлами

Создание файла

%**touch** ***имя_файла***

изменяет время последней модификации файла на текущее. Побочный эффект: если файла нет, он создается с нулевым размером.

%**>*имя_файла***

создает файл нулевого размера (буквально: в этот файл перенаправляется вывод из ниоткуда).

%touch a

%>b

%ls -l

Создание каталога

%mkdir dir

%ls -l

Копирование файла в другой файл или каталог

%**cp** ***файл-источник файл-или-каталог-приемник***

Если файл приемник существует, он будет удален (т.е. копирование производится поверх). Каталог-приемник должен существовать.

%cp b.txt z

%ls -l

UNIX рассчитана на профессиональных пользователей. Предупреждений по поводу удаления файлов не выводится.

Рекурсивное копирование каталога в другой каталог

%**cp -R** ***каталог-источник каталог-приемник***

%cp -R prog dir

%ls -l

%ls -l dir

Перемещение/переименование файла или каталога

%**mv** ***файл-или-каталог-источник файл-или-каталог-приемник***

%mv z y

%ls

%**mv y dir**

%**ls**

%**ls dir**

%**mv dir ddd**

%**ls**

%**ls ddd**

Символические связи

Символическая связь - особый тип файла, содержащий имя другого файла (в листинге *ls -l* такие файлы обозначаются буквой l в первой колонке). Чтение-запись в файл-связь на самом деле приводят к чтению-записи в файл, на который он ссылается.

Создание символической связи:

%**ln -s *сущетсвующий_файл файл_связь***

(Файл с именем ***файл_связь*** не должен существовать.)

%**ln -s b.txt c.txt**

%**ls -l**

(внимание на файл c.txt и его размер: 5 - это число символов в имени "b.txt")

%**more c.txt**

Жесткие связи

В файловой системе UNIX имя файла является указателем на индексный дескриптор, который содержит атрибуты файла и массив адресов дисковых блоков, в которых находятся данные файла. Однако, индексный дескриптор не содержит ссылки на имя файла, следовательно файл может иметь несколько имен. Дескриптор содержит только счетчик числа этих имен, значение которого показывается во второй колонке листинга *ls -l*.

Создание жесткой связи - это создание еще одного имени, ссылающегося на тот же самый индексный дескриптор:

%**ln *сущетсвующее_имя_файла новое_имя_файла***

%**ln b.txt d.txt**

%**ls -l**

(внимание на значение второй колонки для b.txt и d.txt и размер этих файлов, заметьте также время последней модификации)

Файлы b.txt и d.txt абсолютно равноправны, т.к. это два разных имени одного и того же физического файла.

%**touch d.txt**

%**ls -l b.txt d.txt**

(внимание на время последней модификации у обоих файлов)

%**ls -i b.txt d.txt**

(выводит номера индексных дескрипторов - видим, что это один и тот же файл)

Удаление файлов и каталогов

%**rm** ***имя_файла***

- удаляет файл, но не каталог;

%**rmdir** ***имя_каталога***

- удаляет каталог, только если он пуст;

%**rm -r** ***имя_каталога***

- удаляет каталог рекурсивно (т.е. со всеми подкаталогами), но просит подтверждения при удалении файлов, в которые у вас нет права записи (поскольку для удаления файла достаточно иметь право записи в его каталог);

%**rm -rf** ***имя_каталога***

- удаляет каталог рекурсивно и молча.

%ls ddd

%rm ddd/y

%ls ddd

%rm ddd

%rmdir ddd

%rm -r ddd

%ls -l

%rm c.txt

%ls -l b.txt d.txt

%rm d.txt

%rm a b

%ls -l

Удаление файла на самом деле представляет собой уменьшение на 1 счетчика его имен в индексном дескрипторе; соответствующий системный вызов называется unlink(2). Физически файл удаляется системой если он закрыт и если счетчик имен равен нулю. Это свойство используется программистами при создании временных файлов: после создания файла функцией open сразу же вызывается unlink, сбрасывающий счетчик имен в 0. Пока файл открыт, его можно использовать (через файловую переменную, возвращенную функцией open; имени у файла уже нет). После вызова close файл автоматически уничтожается.

СРС - Лабораторная работа "Управление процессами в Unix"

Сценарий: Сбор сведений о системе

В данном сценарии изучаются команды, предоставляющие сведения о системе.

Начальные условия: Командная строка после входа в систему.

1. Определить имя текущей UNIX-системы с помощью команды **uname -a**

user@desktop ~ $ uname -a

Linux thinkpad 2.6.13-gentoo-r4 #1 Mon Oct 17 12:09:36 MSD 2005 i686 Intel(R)...

Здесь выводится полная версия системы, дата компиляции ядра и тип процессора.

2. Вывести содержимое директории proc с помощью команды **ls /proc**:

```
user@desktop ~ $ ls /proc
1/     7/     8113/  8209/  9073/       dri/         modules
103/   7060/  8124/  8211/  9324/       driver/      mounts@
106/   7128/  8125/  8213/  9325/       execdomains  mtrr
185/   7693/  8131/  8215/  937/        fb           net/
186/   7770/  8135/  8319/  9376/       fb1/         partitions
...
```

Признаком операционной системы Linux является наличие директории /proc, которая содержит псевдофайлы, предоставляющие системную информацию и позволяющие настраивать ядро.

3. Вывести текущие пользовательские сеансы с помощью команды **who**:

```
user@desktop ~ $ who
user   tty1         2005-10-24 13:17
user   pts/0        2005-10-24 13:19
user   pts/2        2005-10-24 23:59
```

4. Вывести список всех примонтированных устройств с помощью команды **mount**:

```
user@desktop ~ $ mount
/dev/hda6 on / type reiserfs (rw,noatime)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
udev on /dev type tmpfs (rw,nosuid)
devpts on /dev/pts type devpts (rw)
/dev/hda1 on /mnt/win type ntfs (ro,uid=1000,gid=100,nls=utf8)
```

Первый столбец – имя устройства, второй – точка монтирования, далее следует тип файловой системы и опции монтирования.

5. Вывести загруженность примонтированных дисков с помощью команды **df -h**:

```
user@desktop ~ $ df -h
Файловая система      Разм  Исп  Дост  Исп% смонтирована на
/dev/hda6              24G   21G  3,4G  86% /
udev                  251M  180K  251M   1% /dev
/dev/hda1             9,4G   7,7G   1,7G   83% /mnt/win
none                  251M     0  251M   0% /dev/shm
```

6. Вывести информацию о всех выполняющихся процессах с помощью команды **ps aux**:

```
user@desktop ~ $ ps aux
```

```
USER     PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       1  0.0  0.0   1432   480 ?        S    13:16   0:01 init [3]
root       2  0.0  0.0      0     0 ?        SN   13:16   0:00 [ksoftirqd/0]
root       3  0.0  0.0      0     0 ?        S<   13:16   0:00 [events/0]
..
user    8456  0.3  4.4  34932 22944 ?        S    13:24   2:15 emacs
user   22537  0.0  0.3   3720  1560 pts/3    Ss   22:37   0:00 /bin/bash
          user    8839  0.0  0.1   2644   932 pts/3    R+   23:01   0:00 ps aux
```

7. Оставить в выводе предыдущей команды только системные процессы с помощью команды **ps aux | grep -v user**:

```
user@desktop ~ $ ps aux | grep -v user
USER     PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       1  0.0  0.0   1432   480 ?        S    13:16   0:01 init [3]
root       2  0.0  0.0      0     0 ?        SN   13:16   0:00 [ksoftirqd/0]
root       3  0.0  0.0      0     0 ?        S<   13:16   0:00 [events/0]
...
root    6989  0.0  0.1   1672   704 ?        Ss   Oct25   0:00 /usr/sbin/syslog
root    7128  0.0  0.1   2020   716 ?        Ss   Oct25   0:00 /usr/sbin/fcron
```

Часть процессов, показанные в квадратных скобках, – это потоки ядра, исполняющие специальные функции. Можно выделить демонов – в столбце терминала они имеют символ «?». Например, системный журнал (syslog) или планировщик задач (cron).

8. Вывести иерархию процессов с помощью команды **pstree**:

```
user@desktop ~ $ pstree
init─┬─acpid
     ├─5*[agetty]
     ├─2*[artsd]
     ├─cpufreqd
     ├─dcopserver
     ├─events/0
     ├─fcron
     ├─gconfd-2
     ├─hcid
     ├─kaccess
     ├─kded───kded───kded
     ├─kdeinit─┬─artsd───artsd───artsd
     │         ├─kio_file
     │         ├─kio_imap4───kio_imap4
```

...

В вершине дерева стоит процесс init.

9. Рассмотреть поведение процессов интерактивно с помощью команды **top**.

Сценарий: Управление процессами с помощью сигналов

В данном сценарии изучаются сигналы, управляющие клавиши для передачи процессам сигналов, команды для управления процессами.

Начальные условия: Командная строка после входа в систему.

1. Запустите команду **yes**, производящую бесконечный вывод символа y на экран. Прервите её нажатием **Ctrl+C**.

```
user@desktop ~ $ yes
y
y
y
y
...
y

    user@desktop ~ $
```

При этом запущенному процессу был отправлен сигнал SIGTERM – завершения программы.

2. Запустим сбор информации обо всех файлах системы с помощью команды **find / > files.txt**.

```
user@desktop ~ $ find / > files.txt
```

3. Найдем идентификатор запущенного только что процесса с помощью команды **ps aux | grep find**, запущенной в другом терминале.

```
user@desktop ~ $ ps aux | grep find
user    8178  9.7  0.1   2844  1020 pts/1    R+   23:45   0:00 find /
    user    8230  0.0  0.1   2708   736 pts/2      R+   23:45   0:00 grep find
```

Первая строка содержит необходимый PID.

4. Отправим сигнал завершения этому процессу с помощью команды **kill 8178**, указав в качестве параметара идентификатор процесса.

```
user@desktop ~ $ kill 8178
```

Можно убедиться, что на терминале с запущенной командой появились строка «Terminated» и приглашение.

5. Если попытаться завершить системный процесс, например командой **kill 1**, появится сообщение об ошибке доступа:

```
user@desktop ~ $ kill 1
    bash: kill: (1) - Operation not permitted
```

Отправление сигналов системным процессам может производить только суперпользователь.

6. Альтернативным способом отправления сигналов процессам – по имени процесса, а не по PID – является команда killall. Выполнив команду **killall bash**, мы завершим все командные оболочки, а тем самым и сеансы пользователей.
   user@desktop ~ $ killall bash
7. До этого мы отправляли только сигнал завершения процесса. Он может перехватываться и игнорироваться программами. Неперехватываемым является сигнал SIGKILL, который может быть отправлен, например, следующей командой: **killall -SIGKILL find**
   user@desktop ~ $ killall -SIGKILL find

Сценарий: Выполнение задач в фоновом режиме

В данном сценарии изучается работа с заданиями командной оболочки, запуск заданий в фоновом режиме.

Начальные условия: Командная строка после входа в систему.

1. Запустим длительную команду, например **find / > files.txt** и приостановим её выполнение с помощью нажатия **Ctrl+Z**. При этом процессу посылается сигнал SIGSTOP.

user@desktop ~ $ find / > files.txt

[1]+ Stopped find / > files.txt

user@desktop ~ $

Команда приостановлена и запомнена как задача 1 (номер в квадратных скобках).

2. Текущий список запущенных задач командной оболочки можно посмотреть командой **jobs**.

user@desktop ~ $ jobs

[1]+ Stopped find / > files.txt

3. Возобновить исполнение задания можно командой **fg 1**, аргументом которой является номер задания.

user@desktop ~ $ fg 1

find / > files.txt

4. Если еще раз приостановить процесс, можно запустить задачу в фоновом режиме: **bg 1**.

user@desktop ~ $ fg 1

[1]+ find / > files.txt &

user@desktop ~ $

Тогда можно будет продолжать работу в командной строке.

5. Команды можно сразу запускать в фоновом режиме. При этом необходимо добавить символ «&» (амперсанд) в конец строки команды: **find / -name "*.xml" > xml-list &**

user@desktop ~ $ find / -name "*.xml" > xml-list &

[1] 9413

user@desktop ~ $

При этом командная оболочка выводит номер задания и PID созданного процесса.

Сценарий: Запуск демонов

В данном сценарии рассматриваются демоны как процессы, не связанные ни с одним терминалом.

Начальные условия: Командная строка после входа в систему.

1. Если запустить команду в фоновом режиме и выйти из командной оболочки: например, **find / -name "*.html" -exec grep -Hn "linux loader" \{} \; &** и **exit**, то запущенная команда завершится по сигналу SIGHUP.

user@desktop ~ $ find / -name "*.html" -exec grep -Hn "linux loader"
\{} \; &
[1] 9555
user@desktop ~ $ exit

2. Для того, чтобы программы не получали сигнал SIGHUP, используется специальная команда nohup:**nohup find / -name "*.txt" -exec grep -Hn "linux loader" \{} \; &**

user@desktop ~ $ nohup find / -name "*.html" -exec grep -Hn "linux loader" \{} >> result\; &
[1] 9783
user@desktop ~ $ exit

Эта программа завершится корректно после окончания поиска.

Сценарий: Изменение приоритетов выполняющихся программ

В данном сценарии изучается механизм приоритетов UNIX и команды для изменения приоритетов запускаемых процессов.

Начальные условия: Командная строка после входа в систему.

1. Каждый процесс в системе имеет свой уровень приоритета (в UNIX он называется «nice»), который можно увидеть с помощью команды **ps -l** – в столбце «NI».

user@desktop ~ $ ps -l
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 1000 17223 17222 0 75 0 - 994 wait pts/1 00:00:00 bash
0 R 1000 17485 17223 0 75 0 - 612 - pts/1 00:00:00 ps

По умолчанию, приоритет процессов простого пользователя равен нулю.

2. Для запуска процесса с заданным приоритетом воспользуемся командой nice. Напрмер, запустим создание архива с пониженным приоритетом: **nice -n 10 tar -cjf libraries.tar.bz2 /usr/lib/.**

user@desktop ~ $ nice -n 10 tar -cjf libraries.tar.bz2 /usr/lib/

3. Чтобы изменить приоритет уже запущенной программы используется команда renice. Например, для понижения приоритета заранее запущенного процесса архивации **tar -cjf libraries.tar.bz2 /usr/lib &**воспользуемся командой **renice +10 -p 3442**:

user@desktop ~ $ tar -cjf libraries.tar.bz2 /usr/lib &
[1] 3442
user@desktop ~ $ renice +10 -p 3442

Простые пользователи не могут повышать приоритет процессов, только понижать.

Задания для самоподготовки

1. Сравните время создания корневой директории с текущим временем системы.
2. Получите список всех типов файловых систем, примонтированных в настоящий момент. Какой тип имеет корневая файловая система?
3. Найдите три процесса, использующих самый большой объём памяти. Какой процент памят они потребляют от общего объёма?
4. Получите иерархию родительских процессов для текущей командной оболочки.
5. Найдите, в каком файле в директории /proc хранится информация о системной памяти.
6. С помощью сигнала остановки процесса и команды bg запустите три параллелльных процесса архивации директории /etc, /home и /tmp.

Блок Б1.

1. В Вашем рабочем каталоге находятся файлы song1.txt, song2.txt, song3, newsong, newsong.old , подкаталог songs. Выполнение следующих команд приведет к тому что

cp song* songs _____
cp song*.* songs _____
cp song? songs _____

2. В каком системном каталоге находятся файлы для выполнения основных пользовательских команд, поддерживающихся UNIX?_____
3. Область данный специальных файлов содержит:
    a. Адреса блоков
    b. Мажор, минор
    c. Драйвера.
4. Системные вызовы, это:
    a. Вызовы которые создают, удаляют и используют различные объекты, главные из которых процессы.
    b. Вызовы которые создают, удаляют и используют различные объекты, главные из которых файлы.
    c. Вызовы которые создают, удаляют и используют различные объекты, главные из которых процессы и файлы.
5. В случае, если компоненты ОС являются не самостоятельными модулями, а составными частями одной большой программы образуют :
    a. Монолитное ядро:
    b. Слоенную систему
    c. Гибридное ядро
6. Какой тип файла маркируется буквой « L »
    a. символическая связь
    b. символьное устройство
    c. обычный файл
7. Программное прерывание, это:
    a. Синхронное событие.
    b. Асинхронное событие.
    c. Случайное событие.
8. Событие, возникающее в результате попытки выполнения программой недопустимой команды называется:
    a. Пользовательской ошибкой
    b. Исключительной ситуацией
    c. Системной ошибкой
9. Ядро операционной системы состоит из совокупности программ:

a. Обеспечивающих пользовательский интерфейс и драйвера
b. Обеспечивающих системные и аппаратные прерывания
c. Обеспечивающих пользовательский интерфейс и библиотеки системных вызовов.

10. В рабочем каталоге у Вас есть файлы либо без расширения, либо с расширением dvi. Вам надо очистить весь каталог. Как это сделать?_____
11. Предположим, что у пользователя есть разрешение на чтение из каталога, но нет разрешения на исполнение. Что произойдет, если каталог использовать в качестве параметра команды ls, заданной с опцией "-x"? Что будет, если указана опция "-l"? Поясните свои ответы.

_____

12. Сравните права доступа, которые должны быть у процесса для выполнения следующих действий, и прокомментируйте:

a. Для создания нового файла требуется разрешение на чтение в каталог.
b. Для "редактирования" существующего файла требуется разрешение на исполнение файла.
c. Для удаления связи файла с каталогом требуется разрешение на запись в каталог, а не в файл.
d. Для проведения поиска в каталоге требуется разрешение чтение в каталог.
e. Для удаления каталога требуется разрешение на чтение и исполнение.

13. Обратите внимание на то, что при выполнении команды ls -ld количество связей с каталогом никогда не равно 1. Почему?_____
14. Предположим, что один процесс меняет свой текущий каталог на каталог "/mnt/a/b/c", после чего другой процесс в каталоге "/mnt" монтирует файловую систему. Завершится ли функция mount успешно? Что произойдет, если первый процесс выполнит команду pwd?
15. В чем отличие символов «>>», «>».
16. Как узнать какие пользователи в системе, и к какой группе вы принадлежите.
17. В рабочем каталоге у Вас есть файлы либо без расширения, либо с расширением tex. Вам надо очистить весь каталог от обычных файлов без расширения. Как это сделать?

_____

18. В текущем каталоге есть один файл с именем text.txt. Что выдаст терминал после выполнения следующих команд

ln text.txt today
ln text.txt tommorow
ls

_____

ln -s text.txt today
ln text.txt today
ls

_____

19. В текущем каталоге имеются файлы doc.1txt, doc2.txt, document.txt, mydoc.f77, Monday.txt. Какие результаты при выполнении следующих команд

- ✓ ***ls doc**** _____
- ✓ ***ls doc*.**** _____
- ✓ ***ls doc?*** _____
- ✓ ***rm doc?*** _____
- ✓ ***ls doc*?*** _____
- ✓ ***rm doc*?***_____
- ✓ ***rm doc.*?***_____

20. Вам надо два текстовых файла объединить в один. Как это сделать, не используя редактор?

_____

21. Как заархивировать все файлы, находящиеся в домашнем каталоге, причем с максимальным сжатием. _____
22. Вам надо создать новый каталог new на флешке, при этом дисковод не подмонтирован. Какую команду следует выполнить?_____

23. Вам необходимо напечатать список файлов из текущего каталога, имеющих любые расширения, в имени которых есть буквы a или b, причем как строчные, так и прописные.

_____

24. Вам необходимо напечатать список файлов из текущего каталога, не имеющих расширений, в имени которых есть буквы x и y, причем как строчные, так и прописные.

_____

25. Вам необходимо проверить атрибуты файла prog.for, в частности узнать его собственника. Это можно сделать с помощью команды _____
26. Вам необходимо скопировать все файлы с флешки в текущую директорию на компьютере, работающем под управлением LINUX. Это можно сделать с помощью следующих команд

_____

27. Вам необходимо увидеть полный список файлов, созданных или отредактированных сегодня. Это можно сделать с помощью команды _____
28. Вам нужно освободить место на диске, переполненным старыми файлами. Среди них множество файлов с расширением pdf. Можно ли уничтожить их, не рискуя потерять важную для себя информацию._____
29. Во время работы Вы создали файлы chap1, chap2, chap3, chap10, chap11 и chap12. При наборе следующих команд, Вы получите следующую строку

ls chap?_____

ls chap1?_____

30. Можно ли удалить чужой файл из своего каталога?

_____

31. Что означают права для файла

661_____

744_____

764_____

777 _____

32. Каталог /usr в типичной UNIX-системе применяется для _____

Блок Б2.

Тест по предмету «Структура операционных систем»

ФИО_____

.

1. При помощи какой команды выводится список атрибутов процессов:_____

_____

_____

2. В чем отличие фонового режима от демона. При каком режиме завершение сеанса работы пользователя не приведет к завершению процесса._____

3. При системном вызове задача переходит в режим:

3.1. Не привелигированный

3.2. Привелигированный

3.3. Ассинхронный

4. Системные вызовы это:

4.1. Интерфейс между операционной системой и пользовательской программой.

4.2. Интерфейс между файловой системой и пользовательской программой.

4.3. Интерфейс между операционной системой и файловой системой .

5. Напишите многоразовые операции над процессами:_____

_____

_____

6. Процесс, который запускается при старте системы и не связан ни с одним пользовательским терминалом называется:
6.1. Зомби
6.2. Демон
6.3. Системный

Какой максимальный объем файла может поддерживаться прямой индексацией._____

7. Отсутствие нужной информации в оперативной памяти ведет к:
- 7.1. Временной исключительной ситуации.
- 7.2. Неисправимым исключительным систуациям
- 7.3. Исправимым исключительным ситуациям.

8. Задачи, которые выполняют классические операционные системы:
- 8.1. Планирование заданий и использования процессора;
- 8.2. Включение и выключение компьютера;
- 8.3. Управление аппаратной частью вычислительной системы

9. Внутреннее строение файловой системой имеет следующие функциональные области:
- 9.1. Файл загрузки, суперблок, индексный файл, область данных.
- 9.2. Файл загрузки, суперблок, индексный файл, область данных, область свопинга.
- 9.3. Файл загрузки, индексный файл, область данных, область свопинга.

10. Какими контекстами определяются процессы в ОС Unix:

_____

_____

___

11. Напишите многоразовые операции над процессами:_____

___

12. Индексный дискриптор, это:
- 12.1. Описатель файла
- 12.2. Число указателей на файл
- 12.3. Размер файла
- 12.4. Описатедь процесса

13. Какие функции выполняет подсистема управления процессами:
- 13.1. Создание и уничтожение процессов
- 13.2. Управление вторичной памятью и отображение страниц виртуальной памятью
- 13.3. Планирование процессов
- 13.4. Управление виртуальной памятью процесса
- 13.5. Обеспечение межпроцессного взаимодействия

14. В операционной системе UNIX под процессом-зомби понимают:
- 1.1. процесс, у которого завершился родительский процесс
- 1.2. процесс, находящийся в состоянии закончил исполнение
- 1.3. процесс, который зациклился

15. В операционной системе UNIX исполняющийся процесс может перейти в состояние ожидание:
- 15.1. только из исполнения в режиме пользователя
- 15.2. только из исполнения в режиме ядра
- 15.3. зависит от обстоятельств

16. В каких случаях производится невытесняющее кратковременное планирование процессов?
- 16.1. когда процесс переводится из состояния исполнение в состояние завершил исполнение
- 16.2. когда процесс переводится из состояния исполнение в состояние ожидание
- 16.3. когда процесс переводится из состояния ожидание в состояние готовность

17. Какие из перечисленных ниже компонентов входят в пользовательский контекст процесса?
- 17.1. программный счетчик процесса
- 17.2. информация о всех устройствах ввода-вывода
- 17.3. код и данные в адресном пространстве процесса

18. Какие из перечисленных ниже компонентов входят в регистровый контекст процесса?
- 18.1. программный счетчик процесса
- 18.2. информация о всех устройствах ввода-вывода
- 18.3. код и данные в адресном пространстве процесса

19. Какие из перечисленных ниже компонентов входят в регистровый контекст процесса?
- 19.1. программный счетчик процесса
- 19.2. информация о всех устройствах ввода-вывода
- 19.3. содержимое регистров процессора

20. Какие из перечисленных ниже компонентов входят в системный контекст процесса?
- 20.1. программный счетчик процесса
- 20.2. информация о всех устройствах ввода-вывода
- 20.3. код и данные в адресном пространстве процесса

21. Какие из перечисленных ситуаций возникают предсказуемо?
- 21.1. прерывания
- 21.2. исключительные ситуации
- 21.3. программные прерывания

22. Процессы, которые являются частью ядра, всегда расположены в оперативной памяти, не имеют соответствующих им программ в виде исполняемых файлов и запускаются особым образом при инициализации ядра системы, называются
- 22.1. системные
- 22.2. демоны
- 22.3. прикладные
- 22.4. пользовательские

Приложение 2

ШКАЛА ОЦЕНИВАНИЯ ПИСЬМЕННОГО ОПРОСА (промежуточный контроль – «ЗНАТЬ»)

- Отметкой **(9-10 баллов)** оценивается ответ, который показывает прочные знания основных понятий криптографии, алгоритмов шифрования данных, математических основ применяемых алгоритмов.
- Отметкой **(7-8 баллов)** оценивается ответ, обнаруживающий прочные знания основных понятий криптографии, алгоритмов шифрования данных, математических основ применяемых алгоритмов. Однако допускается одна - две неточности в ответе.
- Отметкой **(4-6 баллов)** оценивается ответ, свидетельствующий в основном о знании основных понятий криптографии, алгоритмов шифрования данных, математических основ применяемых алгоритмов. Допускается несколько ошибок в содержании ответа.
- Отметкой **(2-3 балла)** оценивается ответ, обнаруживающий незнание основных понятий криптографии, алгоритмов шифрования данных, математических основ применяемых алгоритмов. Отмечается отсутствие логичности и последовательности в ответе. Допускаются серьезные ошибки в содержании ответа.
- Отметкой **(0 -1 балл)** оценивается ответ, при котором студент демонстрирует непонимание проблемы или нет ответа.

ШКАЛА ОЦЕНИВАНИЯ ПРАКТИЧЕСКИХ ЗАДАНИЙ (промежуточный контроль – «УМЕТЬ и ВЛАДЕТЬ»)

- Отметкой **(17-20 баллов)** оценивается ответ, при котором студент правильно решает индивидуальные задачи по предложенному алгоритму шифрования, демонстрирует способность грамотно формализовать задачу из индивидуального задания.
- Демонстрирует полное понимание проблемы. Все требования, предъявляемые к заданию, выполнены.
- Отметкой **(11-16 баллов)** оценивается ответ, при котором студент в основном правильно решает индивидуальные задачи по предложенному алгоритму шифрования, демонстрирует способность грамотно формализовать задачу из индивидуального задания. Демонстрирует значительное понимание проблемы. Большинство требований, предъявляемых к заданию, выполнены.
- Отметкой **(4-10 баллов)** оценивается ответ, при котором студент в основном не правильно решает индивидуальные задачи по предложенному алгоритму шифрования, демонстрирует способность грамотно формализовать задачу из индивидуального задания. Демонстрирует частичное или небольшое понимание проблемы. Многие требования, предъявляемые к заданию, не выполнены.

- Отметкой **(0 -3 балла)** оценивается ответ, при котором студент демонстрирует непонимание проблемы или нет ответа и даже не было попытки решить задачу

ШКАЛА ОЦЕНИВАНИЯ РЕФЕРАТА

№	Наименование показателя	Отметка (в %)
1	Во введении четко сформулирован тезис, соответствующий теме реферата, выполнена задача заинтересовать читателя	85 - 100
2	Деление текста на введение, основную часть и заключение	
3	В основной части логично, связно и полно доказывается выдвинутый тезис	
4	Заключение содержит выводы, логично вытекающие из содержания основной части	
5	Все требования, предъявляемые к заданию, выполнены	
6	При защите реферата демонстрирует полное понимание проблемы и для выражения своих мыслей не пользуется упрощенно-примитивным языком.	
1	Во введении четко сформулирован тезис, соответствующий теме реферата, в известной мере выполнена задача заинтересовать читателя	75 – 84
2	В основной части логично, связно, но не достаточно полно доказывается выдвинутый тезис	
3	Заключение содержит выводы, логично вытекающее из содержания основной части	
4	При защите реферата демонстрирует понимание проблемы и для выражения своих мыслей не пользуется упрощенно-примитивным языком.	
1	Во введении тезис сформулирован не четко и не вполне соответствует теме реферата	60 - 74
2	В основной части выдвинутый тезис доказывается недостаточно логично (убедительно) и последовательно	
3	Заключенные выводы не полностью соответствуют содержанию основной части	
4	При защите реферата демонстрирует не полное понимание проблемы и язык работы в целом не соответствует требуемому уровню	
1	Во введении тезис отсутствует или не соответствует теме реферата	40 - 59
2	Деление текста на введение, основную часть и заключение отсутствует	
3	В основной части нет логичного последовательного раскрытия темы	
4	Выводы не вытекают из основной части	
5	При защите реферата демонстрирует полное непонимание проблемы и язык работы можно оценить, как «примитивный».	
1	Работа написана не по теме	менее 58

Раздел 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература			
6.1.1. Основная литература			
	Авторы,	Заглавие	Издательство, год
Л1.1	Таненбаум Э.С.	Современные операционные системы	СПб.:Питер 2015
Л1.2	Гордеев А. В.	Операционные системы: Учебник для вузов	СПб.:Питер 2012
6.1.2. Дополнительная литература			
	Авторы,	Заглавие	Издательство, год
Л2.1	Соловьев Г.Н., Никитин В.Д.	Операционные системы ЭВМ: учебное пособие для вузов	М.: Высшая школа 1989
Л2.2	Олифер В.Г., Олифер	Сетевые операционные системы: Учебник	СПб.: Питер 2002
Л2.3	Партыка Т.Л., Попов	Операционные системы, среды и оболочки: Учебное пособие	М.: ФОРУМ: ИНФРА-М 2004
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"			
Э1	Что такое ОС вообще и Linux в частности		http://www.linuxcenter.ru/lib/books/kostromin/gl_01_01.p

Раздел 5. Методические указания для обучающегося по освоению дисциплины. Математические методы решения прикладных задач.

Текущий контроль осуществляется в течение семестра и включает в себя проверку подготовки студента к выполнению
практической работы (т.е. знание теории по теме практической работы), проверку завершения практической работы и
оформления отчёта.

Рекомендации по организации самостоятельной работы студента

1. Советы по планированию и организации времени, необходимого для изучения дисциплины. Рекомендуется следующим
образом организовать время, необходимое для изучения дисциплины:

Изучение конспекта лекции в тот же день, после лекции – 10-15 минут.

Изучение конспекта лекции за день перед следующей лекцией – 10-15 минут.

Изучение теоретического материала по учебнику и конспекту – 1 час в неделю.

Подготовка к практическому занятию – 2 час.

Всего в неделю – 3 часа 30 минут.

2. Описание последовательности действий студента

Для понимания материала и качественного его усвоения рекомендуется такая последовательность действий:

2.1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно
сначала просмотреть и обдумать текст лекции, прослушанной сегодня (10-15 минут).

2.2. При подготовке к лекции следующего дня, нужно просмотреть текст предыдущей лекции, подумать о том, какая может
быть тема следующей лекции (10-15 минут).

2.3. В течение недели выбрать время (1-час) для работы с рекомендуемой литературой в библиотеке.

2.4. При подготовке к практическим занятиям следующего дня, необходимо сначала прочитать основные понятия и подходы
по теме домашнего задания. При выполнении упражнения или задачи нужно сначала понять, что требуется в задаче, какой
теоретический материал нужно использовать, наметить план решения задачи.

3. Рекомендации по использованию материалов учебно-методического комплекса. Рекомендуется использовать
методические указания по курсу , текст лекций преподавателя.

4. Рекомендации по работе с литературой. Теоретический материал курса становится более понятным, когда дополнительно
к прослушиванию лекции и изучению конспекта, изучаются и книги. Легче освоить курс, придерживаясь одного учебника и
конспекта. Рекомендуется, кроме «заучивания» материала, добиться состояния понимания изучаемой

темы дисциплины. С
этой целью рекомендуется после изучения очередного параграфа выполнить несколько простых упражнений на данную
тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот
параграф?, какие новые понятия введены, каков их смысл?, что даст это на практике?.

5. Советы по подготовке к рубежному и промежуточному контролям. Дополнительно к изучению конспектов лекции
необходимо пользоваться учебником. Кроме «заучивания» материала, очень важно добиться состояния понимания
изучаемых тем дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько
упражнений на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить
на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?, что даст это на практике?.
При подготовке к промежуточному контролю нужно изучить теорию: определения всех понятий и подходы к оцениванию
до состояния понимания материала и самостоятельно решить несколько типовых задач из каждой темы. При решении задач
всегда необходимо уметь качественно интерпретировать итог решения.

6. Указания по организации работы с контрольно-измерительными материалами, по выполнению домашних заданий. При
выполнении домашних заданий необходимо сначала прочитать основные понятия и подходы по теме задания. При
выполнении упражнения или задачи нужно сначала понять, что требуется в задаче, какой теоретический материал нужно
использовать, наметить план решения задачи, а затем приступить к расчетам и сделать качественный вывод.
Рубежный контроль осуществляется в течение семестра в виде защиты практических работ, выполнения контрольных
работ, тестирования и контрольных вопросов.
Методические указания по выполнению практических работ представлены в электронной папке преподавателя (локальная
сеть кафедры Информационных и вычислительных технологий КРСУ). Выполнение практических работ завершается
оформлением отчета, в котором приводится теория по теме работы, результаты работы программы и графики.
Рекомендации по работе с тестами по дисциплине.
Совокупность тестов , представленная в разделе 5.3 может использоваться по частям для текущего контроля освоения
предмета по разделам и темам и для итогового контроля знаний по дисциплине в целом.
Предполагается «загрузить» тесты в специализированную ПС или использовать их в форме бланочного контроля. Они
могут использоваться студентами для самоконтроля при подготовке к итоговому зачёту по дисциплине.
Вопросы по контрольным работам представлены в разделе РПД Фонд оценочных средств.
Ответы на контрольные работы оформляются в письменном виде.
Список литературы должен быть составлен в полном соответствие с действующим стандартом (правилами).
Для допуска к зачету студент в течении семестра должен сдать все практические работы.