**Вопросы курсовая**

1. **Имя компьютера.**

$ hostname

1. **Имя пользователя.**

$ whoami

1. **Информация о пользователях ОС;**

$ w

1. **Версию операционной системы.**

$ uname –a

$ cat /src/\*-version

-a – вывести всю доступную информацию о системе

1. **Информация о видеокарте.**

$ lspci | grep -i vga

lspci – информация о подключенных PCI-устройствах

grep -i – фильтр, игнорирующий регистр (-i)

1. **Информация об устройстве ПЗУ.**

*Постоянное запоминающее устройство (ПЗУ) — это энергонезависимая память, которая используется для хранения массива неизменяемых данных.*

$ free –h – mem (физ память) это как раз таки пзу

1. **Информация о звуковой карте.**

$ lspci | grep -i audio

1. **Информация об устройствах ввода/вывода.**

$ lspci – вывод всех устройств, подключенных к шине PCI

(мб $ lsinput и $ xinput, но они не стандартные вроде как)

1. **Имя процесса приложения.**

$ ps –p <PID>

1. **Время работы выбранного процесса.**

$ ps -o etime <PID>

-o - output т.е какие столбцы мы выводим.

etime – время работы процесса

1. **Время работы ОС.**

$ uptime

1. **Процент используемой физической памяти.**

$ free -m | awk 'NR==2{printf "%.2f%\n", $3\*100/$2}'

free –m - выводит информацию о памяти в мегабайтах. Результат содержит несколько строк, каждая из которых представляет различные аспекты использования памяти, такие как общий объем, использованный, доступный и т. д. Этот вывод передается через конвейер (pipe) оператор | на следующую команду.

awk 'NR==2{printf "%.2f%\n", $3\*100/$2 }' - Эта часть команды использует awk, инструмент для обработки текстовых данных, чтобы выполнить следующее:

NR==2 – условие, которое указывает awk обработать только вторую строку вывода free –m, т.к. вторая строка содержит информацию о физической памяти.

{printf "%.2f%\n", $3\*100/$2 } – действие awk. Оно берет третий столбец, который содержит количество используемой памяти, умножает его на 100, затем делит на второй столбец, который содержит общий объем памяти. Результат форматируется с помощью printf в виде процента с двумя знаками после запятой (%.2f) и выводится на экран.

1. **Процент используемой виртуальной памяти.**

$ free -m | awk 'NR==3{printf "%.2f%\n", $3\*100/$2}' – пояснение смотреть в 12 вопросе.

1. **PID процесса.**

*Идентификатор процесса (PID) — это уникальный номер (идентификатор) процесса в многозадачной операционной системе.*

$ pgrep <название\_приложения>

1. **Приоритет процесса.**

В Linux любой процесс может иметь приоритет от -20 до +19. Максимальным приоритетом считается тот процесс, у которого минимальное число. Например, максимальный по приоритету процесс будет иметь число -20, а минимальный -19.

Задача с приоритетом -20 будет выполняться в первую очередь с максимальным использованием ресурсов, а задача с +19 — в последнюю очередь с минимальным использованием ресурсов.

Для установки приоритетов в Linux используются программы nice и renice.

$ ps –o nice <PID>

-o - output т.е какие столбцы мы выводим

nice – значение приоритета процесса

1. **Пользователь выбранного процесса.**

$ ps -o user <PID>

user – вывод пользователя процесса

1. **Количество пользовательских процессов.**

$ ps -u <имя\_пользователя> | wc –l

-u – вывод процессов, принадлежащих пользователю

wc –l – подсчет количества строк

1. **Количество системных процессов.**

$ ps -eo stat= | grep '^S' | wc –l

ps -eo stat= - выводит только статусы процессов

e – все процессы

grep '^S' – фильтр того, что строки начинаются с S. В контексте ps, S означает, что процесс является системным

wc –l – подсчет количества строк

1. **Всего процессов.**

$ ps -e | wc –l

wc –l – подсчет количества строк

1. **Всего потоков.**

$ ps -eL | wc –l

-L – флаг для вывода потоков

1. **Информация о статусе беспроводной сети.**

$ iwconfig

1. **Информация о сетевых настройках.**

$ ifconfig

1. **Продолжительность текущего сеанса работы пользователя.**

$ who -u | awk '{print $5}'

awk – инструмент для обработки текстовых данных. Выводит 1 и 5 столбцы, которые содержат имя пользователя, время входа и время активности

1. **Информация о процессоре.**

$ lscpu

1. **Загрузка каждого ядра ЦП в %.**

$ mpstat -P ALL

-P ALL – вывод статистики для всех ядер процессора

Последний столбец %idle – показывает процент времени, когда ядро было бездействующим, что можно рассматривать как загрузку процессора. Значение 100% в этом столбце означает, что ядро не выполняло никаких задач, а значение 0% - что ядро было полностью загружено

1. **Общая загрузка процессора в %.**

$ mpstat

Загрузку смотрим так же, как и в прошлом вопросе

1. **Размер файла подкачки в байтах.**

$ free -b | awk 'NR==3 {print $2}'

1. **Количество свободных байтов файла подкачки.**

$ free -b | awk 'NR==3 {print $4}'

1. **Значение загрузки процессора данным процессом.**

$ ps -o %cpu <PID >

1. **Количество модулей, используемых процессом.**

ls /proc/<pid>/maps | grep "\.ko"

1. **Размер рабочего множества страниц.**

$ pmap -x <pid> | tail -n 1 | awk '{print $4}'

pmap -x – вывод детальной информации о памяти, используемой процессом

tail –n 1 – оставляет только последние (1 в данном случае) строки

1. **Счетчик количества созданных описателей (дескрипторов).**

$ ls /proc/<pid>/fd/ | wc –l

ls – вывод содержимого каталога

1. **Маска привязки процесса к процессорам.**

$ taskset -p <pid>

1. **Количество страничных ошибок.**

ps -o min\_flt,maj\_flt <pid> | awk 'NR==2 {printf"%d", $1+$2}'

1. **Время старта процесса (час: мин: сек).**

$ ps –o lstart <pid> | awk '{print $4}'

1. **Количество потоков процесса.**

$ ps -T <pid> | wc –l

-T – флаг для отображения информации о потоках

1. **Значение размера используемой оперативной памяти.**

$ free -h | awk 'NR==2 {print $3}'

1. **Пиковое значение размера используемой виртуальной памяти.**

$ grep Vm /proc/<pid>/status | grep -i vmpeak

1. **Оперативная память.**

Оперативная память (RAM) - это вид компьютерной памяти, который используется операционной системой для временного хранения данных, с которыми в настоящее время работает процессор. Она обеспечивает доступ к данным намного быстрее, чем жесткий диск, но её содержимое теряется при выключении компьютера.

1. **Определение системных утилит.**

Системные утилиты - программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики, тестирования аппаратных и программных средств, оптимизации использования дискового пространства, восстановления разрушенной на магнитном диске информации и т.п.). В Linux это Терминал, Управление компьютером, Монитор ресурсов и т.д.

1. **Понятие лог-файла.**

Лог-файл - это файл, который содержит записи событий или сообщений, создаваемых программами или операционной системой в процессе их работы. Логи используются для отслеживания действий, обнаружения проблем и анализа производительности системы.

1. **Тестирование программного обеспечения.**

Тестирование программного обеспечения (ПО) - это процесс выполнения программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор данных называется тестовым или просто тестом

1. **Стратегия тестирования.**

Стратегия тестирования - это план проведения работ по тестированию системы или её модуля, учитывающий специфику функциональности и зависимости с другими компонентами системы и платформы. Стратегия определяет типы тестов, которые нужно выполнять для данного функционала системы, включает описание необходимых подходов с точки зрения целей тестирования.

Стратегия тестирования отвечает на вопросы:

* как, каким образом тестирование даст ответ, что данный функционал работает;
* что нужно сделать и чем пользоваться из инструментальных средств, для достижения целей тестирования;
* когда определённый функционал будет тестироваться и когда ожидать получения результатов.

1. **Файловые менеджеры.**

Файловые менеджеры - это программы, предназначенные для управления файлами и папками на компьютере. Они позволяют пользователю просматривать, копировать, перемещать, удалять и в прочем обрабатывать файлы и папки в операционной системе. Примеры таких программ включают в себя проводник Windows, Finder на macOS и Nautilus в среде Linux.

1. **Понятие Операционная система.**

Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

1. **Файловые системы. Понятие и описание.**

Файловая система - это способ организации и хранения данных на носителе информации, таком как жесткий диск или флэш-накопитель. Она определяет структуру файлов, способы их доступа, управления и навигации по файлам и папкам. Файловые системы могут различаться по типу, например, FAT32, NTFS, ext4 и другие, и каждая из них имеет свои особенности и возможности.

1. **Межпроцессное взаимодействие - Отображение файлов.**

Межпроцессное взаимодействие (IPC) - Отображение файлов (File Mapping) - это метод IPC, который позволяет нескольким процессам разделять область памяти через отображение файла в адресное пространство каждого процесса. Это позволяет процессам обмениваться данными, работая с общей областью памяти, что может быть полезно для совместного доступа к данным или совместной работы над файлами.

1. **Межпроцессное взаимодействие - Сокеты.**

Межпроцессное взаимодействие (IPC) - Сокеты (Sockets) - это конечная точка для обмена данными между процессами через сеть или между процессами на одном компьютере. Сокеты могут быть использованы для передачи данных по протоколам TCP/IP или UDP/IP, обеспечивая коммуникацию между процессами на разных компьютерах или на одном компьютере через сеть.

1. **Межпроцессное взаимодействие - Каналы.**

Межпроцессное взаимодействие (IPC) - Каналы (Pipes) - это метод IPC, который обеспечивает однонаправленный поток данных между процессами в операционной системе. Каналы могут быть анонимными или именованными и используются для передачи данных от одного процесса к другому, например, вывод одного процесса может быть направлен на вход другого процесса через канал.

1. **Межпроцессное взаимодействие - Очередь сообщений.**

Межпроцессное взаимодействие (IPC) - Очередь сообщений (Message Queues) - это метод IPC, который позволяет процессам обмениваться сообщениями через общую очередь в операционной системе. Процессы могут помещать сообщения в очередь и извлекать их из неё, обеспечивая асинхронную коммуникацию и синхронизацию между процессами.

1. **Межпроцессное взаимодействие - Разделяемая память.**

Межпроцессное взаимодействие (IPC) - Разделяемая память (Shared Memory) - это метод IPC, который позволяет нескольким процессам обмениваться данными, используя общую область оперативной памяти. Процессы могут читать и записывать данные в эту область памяти, обеспечивая быстрый и эффективный обмен данными без необходимости использования файлов или сокетов.

1. **Иерархия каталогов и файловых систем.**

Иерархия каталогов и файловых систем - это способ организации файлов и папок на компьютере, представленный в виде древовидной структуры. Каталоги могут содержать файлы и другие каталоги, образуя иерархию, которая позволяет пользователю легко организовывать и находить нужные файлы.

1. **Свойства файлов, папок и дисков.**

Свойства файлов, папок и дисков могут включать в себя различные атрибуты и параметры, такие как права доступа (для файлов и папок), размер, дата создания, дата последнего изменения, тип (для файлов), метка тома (для дисков), объём доступного пространства и другие характеристики, которые определяют их поведение и свойства в системе.

1. **Директория. Понятие и описание.**

Директория - это особый тип файла, который содержит список файлов и подкаталогов в определенной области файловой системы. Директории используются для организации и структурирования файлов и папок в файловой системе, обеспечивая удобное и логическое разделение данных.

1. **Понятие Процесс. Описание и назначение процессов.**

Процесс – это экземпляр программы, выполняющейся в операционной системе. Каждый процесс имеет свою собственную область памяти, в которой хранятся данные и инструкции программы. Кроме того, каждый процесс имеет свой собственный набор ресурсов, таких как файлы, сокеты и т.д. Процессы изолированы друг от друга и не могут напрямую обмениваться данными. Для обмена данными между процессами используются механизмы межпроцессного взаимодействия (IPC).

Назначение: Процессы обеспечивают изоляцию и управление ресурсами на уровне программы.

1. **Понятие Поток. Описание и назначение потоков.**

Поток – это легковесный процесс, который выполняется в рамках одного процесса. Каждый поток имеет свой собственный стек вызовов и может выполняться параллельно с другими потоками в пределах процесса. Все потоки процесса имеют общую область памяти и могут обмениваться данными между собой без использования механизмов межпроцессного взаимодействия.

Назначение: Потоки позволяют использовать многопоточное программирование, которое может улучшить производительность программы.

1. **Понятие Многопоточность. Описание и назначение потоков.**

Многопоточность - это концепция, при которой программа состоит из нескольких потоков исполнения, которые работают параллельно или асинхронно. Многопоточность позволяет программе эффективно использовать многоядерные процессоры, улучшает отзывчивость приложений и обеспечивает более эффективное использование ресурсов компьютера.

1. **Логическая организация механизма передачи информации.**

Логическая организация механизма передачи информации - это способы организации и обработки данных в компьютерных системах с точки зрения программного и логического уровней. Это включает в себя методы кодирования, сжатия, шифрования и передачи данных между компонентами системы, такими как процессоры, память, устройства ввода-вывода и сети.

1. **Организация памяти.**

Организация памяти - это способы управления и структурирования доступа к физической памяти компьютера, такой как оперативная память (RAM) и виртуальная память. Организация памяти включает в себя методы выделения, освобождения и управления блоками памяти, а также механизмы виртуальной памяти, которые позволяют программам использовать больше памяти, чем доступно физически.

1. **Физическое и логическое адресные пространства.**

Физическое и логическое адресные пространства - это способы организации адресов памяти в компьютерной системе. Физическое адресное пространство представляет собой адреса физических ячеек памяти, а логическое адресное пространство - адреса, используемые программами и операционной системой для доступа к памяти. Процессор выполняет преобразование логических адресов в физические с помощью механизмов управления памятью, таких как таблицы страниц или кэши.

1. **Понятие о виртуальной памяти.**

Виртуальная память - это механизм, используемый операционной системой для управления доступом к памяти компьютера. Он позволяет программам использовать больше памяти, чем физически доступно, путем создания виртуального адресного пространства для каждого процесса. Виртуальная память включает в себя использование страницирования и сегментации для управления памятью и обеспечения безопасности и изоляции процессов.

1. **Принцип адресации.**

Принцип адресации - это метод обращения к памяти компьютера путем указания адреса нужного байта или блока памяти. Адресация позволяет программам и операционной системе находить и обрабатывать данные и инструкции в памяти компьютера, используя различные способы кодирования адресов и доступа к ним.

1. **Способы организации виртуальной памяти.**

Способы организации виртуальной памяти включают в себя страничное и сегментное разбиение. Страничное разбиение разделяет виртуальную память на страницы фиксированного размера, которые загружаются и выгружаются из физической памяти по мере необходимости. Сегментное разбиение разделяет виртуальную память на логические сегменты переменной длины, каждый из которых может быть загружен в физическую память отдельно.

1. **Системные вызовы для управления файлами.**

open(), read(), write(), close(), rename(), chmod()

1. **Системные вызовы для управления каталогами.**

opendir(), readdir(), closedir(), mkdir(), rmdir()

1. **Системные вызовы для процессов.**

exec(), wait(), getpid(), kill(), exit()