**Вопросы по 5 занятию**

1. Какие типы процессов бывают в юниксе, и чем они отличаются?

**Пользовательские (прикладные) процессы** – процессы, имеют управляющий терминал, запущенные пользователем в терминале, ограничены сеансом работы пользователя, возможно выполнение в фоновом режиме.

**Демоны (daemon)** – работают в фоновом режиме, запускаются при старте системы, выполняют системные задачи и сервисы, не имеют управляющего терминала, родительский pid обычно 1, название кончается на d.

**Системные процессы** - являются частью ядра, всегда расположены в оперативной памяти, не имеют соответствующих исполняемых файлов, запускаются особым образом при инициализации ядра.

2. Какие варианты запуска скриптов есть?

Обычный запуск (в mc или командной строке)

В текущем Shell – *. command*

В фоновом режиме – *command &*

По расписанию */etc/crontab*

В будущем – *at –t* ВВ][ГГ]ММДДччмм[сс] команда

Через Screen

3. Как посмотреть список процессов?

**top** – для отображения процесса, который потребляет наибольшее количество ресурсов сервера

ps – список процессов обычно используется с разными ключами или в скриптах. (ps aux расширенный список)

4. Как найти и снять нужный процесс?

*ps –A* - полный список процессов, или *top |grep* *имя нужного* процесса

**kill -9 PID (**1 = Hung up, 9 = Kill, 15 = Terminate)

pkill имя процесса (убить процесс по имени)

killall имя процесса (убить процесс и все его дочерние процессы)

5. Как посмотреть самые прожорливые процессы?

*top*

6. Какую информацию можно получить из команды top?

Pid – номер процесса

User – пользователь процесса

Pr и Ni(nice) приоритет в данный момент и приоритет, присвоенный командой nice.

S — текущий статус процесса: R — running; S — sleeping, Z — zombie

%CPU — процент используемого времени центрального процессора

\*\*%MEM — процент ОЗУ, используемой процессом

\*\*TIME+ — продолжительность работы процесса с момента запуска

\*\*COMMAND — название команды (программы), которая инициировала процесс.

Первая строчка (top) дает нам общее представление о загрузке системы. Здесь указано:

\*\* текущее время

\*\* up — продолжительность работы системы с момента запуска

\*\* user — текущее количество пользователей системы

\*\* load average — средняя загруженность системы одну минуту назад, пять и 15 соответственно.

7. Как запустить процесс в фоновом режиме?

**Нужно добавить в конце *&***

8. Как запустить процесс через screen?

*Screen*

Команда **screen** является эффективным инструментом для системного администрирования, особенно при работе с системой без среды рабочего стола. Имея один терминал, вы можете создавать несколько окон для обработки параллельных процессов без конфликтов между ними.

9. Как запустить процесс через cron?

systemctl status crond.service – проверить запущен ли крон

etc/crontab

/2 шаг=2

- диапазон (range of values)

, (value list separator) список разделенных значений

\* любой

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Minute | Hour | Day of month (1-31) | Month | Day of week |
| \* | \* | \* | \* | \* |
| \*/2 |  |  |  |  |

10. Что такое позиционные параметры?

$0 - имя скрипта

$1,$2 - первый, второй позиционный параметр

$# - количество позиционных параметров

$\*, $@ - все позиционные параметры

$? - код завершения последней операции

$$ - PID скрипта

$! - PID последнего процесса в фоновом режиме

Позиционные параметры – зарезервированные системой переменные, которые указываются после команды,

Скрипт принимает параметры, числа или строки, указанные после скрипта, при запуске скрипта на выполнение через командную строку.

11. Что делают двойные кавычки?

“ “ - не отменяет спец символы

12. Что делают одинарные кавычки?

‘ ‘ - отменяет спец символы

13. Что делают обратные кавычки?

` ` - для выполнения команд

14. Чем отличаются разные виды скобок?

( ) - запуск нескольких команд в отдельном shell

{ } - используется для группировки данных

[ ] - проверка условий

15. Как создать и использовать функции?

**имя\_функции() { список\_команд }**Имя функции не должно совпадать ни с одной из существующих команд или функций, а все команды в теле функции пишутся с новой строки

Функция запускается так же, как и файл.  ./function.sh

16. Как использовать арифметические функции?

Необходимо сообщить в переменные их значения для вычисления, если количество переменных меньше или больше, выводить сообщение с помощью условий.

17. Как использовать отладчик скриптов?

bash –n скрипт – Не выполняет скрипт, а только проверяет синтаксис

set –v - Командная оболочка печатает входные строки сразу, как они считываются.

set –x - Перед исполнением команды выдаются трассировочные данные.

18. Что означает в первой строчке конструкция: #!/bin/sh?

Используем интерпретатор **sh (bourne shell)**