*Лабораторная работа №6*

**Мониторинг процессов в операционной системе Linux**

**Цель работы:** приобретение навыков получения информации о запушенных процессах и используемых ими ресурсах в операционной системе Linux

**1. Методические рекомендации**

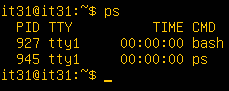
**Идентификация процессов**

Cистема идентифицирует процессы по уникальному номеру, называемому идентификатором процесса или PID (process ID). Все процессы, работающие в системе GNU/Linux, организованы в виде дерева. Корнем этого дерева является init – процесс системного уровня, запускаемый во время загрузки. Для каждого процесса хранится идентификатор его родительского процесса (PPID, Parent Process ID). У процесса init PPID равен 0.

**Получение общих сведений о запущенных процессах**

***Команда ps (сокращение от process status)***

Запуск ps без аргументов покажет только те процессы, которые были запущены Вами и привязаны к используемому Вами терминалу.



Часто используемые параметры (указываются без "-"):

a – вывод процессов, запущенные всеми пользователями;

x – вывод процессов без управляющего терминала или с управляющим терминалом, но отличающимся от используемого Вами;

u – вывод для каждого из процессов имя запустившего его пользователя и времени запуска.



Обозначения колонок в типовом выводе команды ps:

PID, PPID – идентификатор процесса и его родителя.

%CPU – доля процессорного времени, выделенная процессу.

%MEM – процент используемой оперативной памяти.

VSZ – виртуальный размер процесса.

TTY – управляющий терминал, из которого запущен процесс.

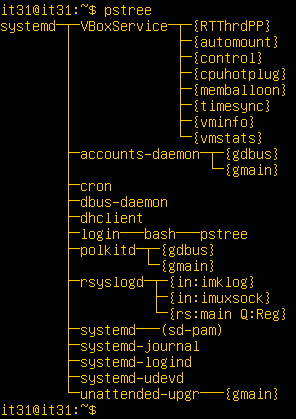
STAT – статус процесса: R – процесс выполняется в данный момент S – процесс ожидает (т.е. спит менее 20 секунд) I – процесс бездействует (т.е. спит больше 20 секунд) D – процесс ожидает ввода/вывода (или другого недолгого события), непрырываемый Z – zombie-процесс T – процесс остановлен

START – время запуска процесса.

TIME – время исполнения на процессоре.

***Команда pstree***

Команда pstree выводит процессы в форме дерева: можно сразу увидеть родительские процессы.



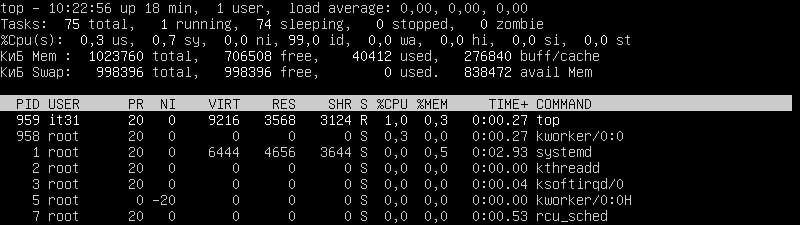
Часто используемые параметры:

-p – вывод PID всех процессов

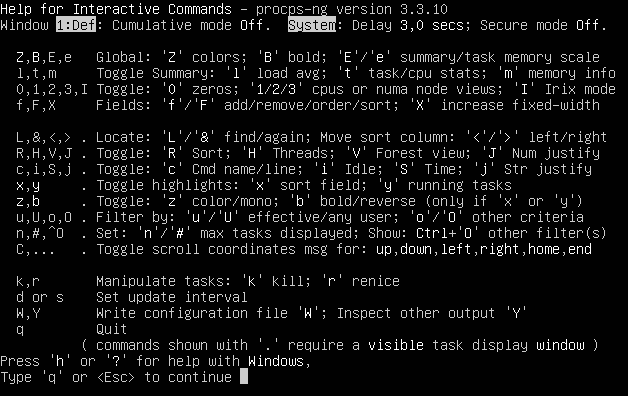
-u – вывод имени пользователя, запустившего процесс.

***Команда top***

top – программа, используемая для наблюдения за процессами в режиме реального времени.



Полностью управляется с клавиатуры. Вы можете получить справку, нажав на клавишу h.



Наиболее полезные команды для мониторинга процессов:

Shift+M – эта команда используется для сортировки процессов по объему занятой ими памяти (поле %MEM);

Shift+P – эта команда используется для сортировки процессов по занятому ими процессорному времени (поле %CPU). Это метод сортировки по умолчанию;

U – эта команда используется для вывода процессов заданного пользователя. top спросит у вас его имя. Вам необходимо ввести имя пользователя, а не его UID. Если вы не введете никакого имени, будут показаны все процессы;

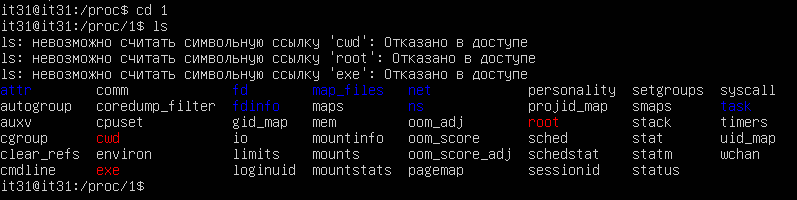
i – по умолчанию выводятся все процессы, даже спящие. Эта команда обеспечивает вывод информации только о работающих в данный момент процессах (процессы, у которых поле STAT имеет значение R, Running). Повторное использование этой команды вернет Вас назад к списку всех процессов.

**Получение детальных сведений о запущенных процессах**

***/proc*** – псевдо-файловая система, которая используется в качестве интерфейса к структурам данных в ядре.



Большинство расположенных в ней файлов доступны только для чтения, но некоторые файлы позволяют изменять переменные ядра. Каждому запущенному процессу соответствует подкаталог с именем, соответствующим идентификатору этого процесса (его PID). Каждый из этих подкаталогов содержит псевдо-файлы и каталоги.



Часть из этих файлов доступна только в директориях процессов, запущенных от имени данного пользователя или при обращении от имени root.

Наиболее часто использующиеся для мониторинга процессов псевдо-файлы и каталоги:

***cmdline*** – файл, содержащий полную командную строку запуска процесса.



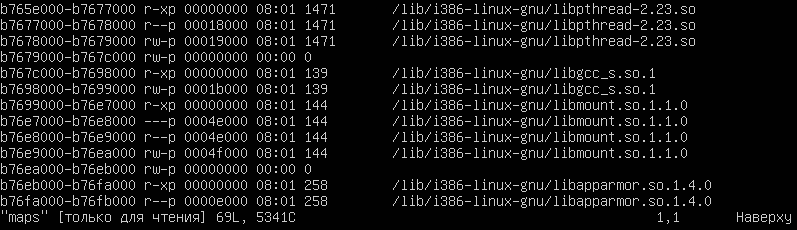
***cwd*** – ссылка на текущий рабочий каталог процесса.

***environ*** – файл, содержащий окружение процесса. Записи в файле разделяются нулевыми символами, и в конце файла также может быть нулевой символ.

***exe*** – символьная ссылка, содержащая фактическое полное имя выполняемого файла.

***fd*** – подкаталог, содержащий одну запись на каждый файл, который в данный момент открыт процессом. Имя каждой такой записи соответствует номеру файлового дескриптора и является символьной ссылкой на реальный файл. Так, 0 – это стандартный ввод, 1 – стандартный вывод, 2 – стандартный вывод ошибок и т. д.

***maps*** – файл, содержащий адреса областей памяти, которые используются программой в данный момент, и права доступа к ним. Формат файла следующий:



где столбцы

1. address - адресное пространство, занятое процессом;

2. perms - права доступа к нему:

r = можно читать

w = можно писать

x = можно выполнять

s = можно использовать несколькими процессами совместно

p = личная (копирование при записи);

3. offset - смещение в файле,

4. dev - устройство (старший номер : младший номер);

5. inode - индексный дескриптор на данном устройстве: 0 означает, что с данной областью памяти не ассоциированы индексные дескрипторы;

6. stat – детальная информация о процессе в виде набора полей.

***sched*** – предоставляет информацию о процессе, использующуюся планировщиком задач.

***statm*** – предоставляет информацию о состоянии памяти в страницах как единицах измерения.



Список полей в файле:

1. size общий размер программы

2. resident размер резидентной части

3. share разделяемые страницы

4. trs текст (код)

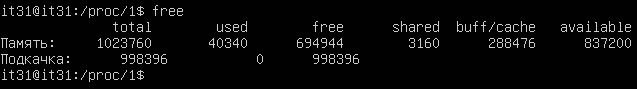
5. drs данные/стек

6. lrs библиотека

7. dt "дикие" (dirty) страницы

**Получение данных об оперативной памяти**

***free*** - возвращает информацию о свободной и используемой памяти в системе, как физической, так и виртуальной (в разделе подкачки на жестком диске).



Поля вывода команды:

total – общее количество доступной физической памяти. Некоторая область оперативной памяти может быть зарезервирована ядром, поэтому показатель total может быть меньше реального объема оперативной памяти.

used – объем используемой памяти (used=total-free).

free - свободная память.

shared - память, распределенная между процессами.

buffers - память используемая в буферах.

cached - память используемая для кэширования.

buffers/cache - использованная память без учета буферов и кэшей/свободная память с учётом буферов и КЭШей.

**Обработка данных о процессах**

Обработка данных о процессах проводится, как правило, в рамках организации конвейера команд обработки текстовых потоков и (или) через циклическую обработку строк файлов. Используют команды, изученные в рамках предыдущей лабораторной работы: ***grep***, ***sed***, ***awk***, ***tr***, ***sort***, ***uniq***, ***wc***, ***paste***, а также функции для работы со строками.

**2. Задания для самостоятельной работы**

1. Изучите команды, рассматриваемые в методических рекомендациях. Вставьте в отчет скрин-шоты их запуска. При этом опишите полученные результаты.

2. Напишите скрипт, решающий задачу вывода в файл списка PID всех процессов, которые находятся в статусе ожидания. Опишите работу скрипта.