

Лабораторная работа 4

ЗНАКОМСТВО С ПРОЦЕССАМИ

Цель работы

Познакомиться с понятием процесса. Научиться получать список имеющихся в системе процессов и управлять их состоянием.

1. Теоретические сведения

1.1. Понятие процесса

Операционная система Linux является *многозадачной (мультизадачной)*. Это значит, что одновременно в системе может присутствовать множество процессов, каждому из которых доступно определенное количество процессорного времени. Для пользователя создается «иллюзия» одновременного выполнения процессов.

Процесс – выполняемая программа с её данными и контекстом.

Каждый процесс имеет уникальный в любой момент времени номер в системе – PID. Первый запускающийся в системе процесс `init`, имеет `pid = 1`.

Для описания процессов в операционной системе имеется список структур – *дескрипторов*, содержащих информацию об идентификаторе процесса, приоритете, состоянии процесса, информацию о принадлежности пользователю и группе, занимаемых процессом ресурсах и др.

Каждый процесс в системе Linux запускается каким-либо процессом. Запускающий процесс – *родительский*, новый процесс – *дочерний*. Процессы, выполняющие одну задачу, объединяются в группы, имеющие собственный идентификатор. Процесс внутри группы, идентификатор которого совпадает с идентификатором группы процессов, считается *лидером группы процессов*.

Все запущенные процессы условно (в зависимости от выполняемой ими функции) можно разделить на три типа:

Системные процессы являются частью ядра и всегда расположены в оперативной памяти. Они часто не имеют соответствующих им программ в виде исполняемых файлов и всегда запускаются особым образом при загрузке ядра системы.

Процессы-демоны – это неинтерактивные процессы, которые выполняются в *фоновом режиме*.

К *прикладным* относятся все остальные процессы, выполняющиеся в системе.

Интерактивные процессы связаны с определённым терминалом и через него взаимодействуют с пользователем. *Фоновые* процессы выполняются независимо от пользователя и (псевдо)параллельно.

Каждый процесс в операционной системе Linux может находиться в одном из четырёх состояний: *работоспособный*, *спящий* (или ожидающий), *остановленный* и *завершившийся*.

1.2. Дополнительные утилиты

Для получения информации о запущенных процессах часто используется команда **ps**. Вывод запущенной без аргументов команды содержит: информацию о процессах текущего пользователя и ассоциированных с текущим терминалом, процессорное время, занятое этим процессом, и имя исполняемого файла. Управлять форматом вывода можно с помощью дополнительных опций (Таблица 1).

\$ ps

Таблица 1. Примеры опций команды ps

Ключ	Описание
-a	Выдать все процессы системы, включая лидеров сеансов.
-d	Выдать все процессы системы, исключая лидеров сеансов.
-e	Выдать все процессы системы.
-x	Выдать процессы системы, не имеющие контрольного терминала.
-o	Определяет формат вывода в виде списка полей, разделенных символом «,».
-u	Выдать процессы, принадлежащие указанному пользователю.

Например, можно получить выборочную информацию обо всех процессах в системе:

\$ ps -eo s,pid,tty,command

Альтернативным способом узнать о состоянии процессов в реальном времени является использование команды **top**. Выводом команды можно управлять с помощью специальных комбинаций клавиш. Справочную информацию можно получить, нажав клавишу «h».

Чтобы получить информацию о запущенных в системе процессах в виде дерева, можно использовать утилиту **pstree**.

2 Управление процессами

Чтобы запустить программу достаточно ввести ее имя в командной строке и нажать «**Enter**». Однако не все команды запускают единственный процесс.

Интерактивные процессы, запущенные в терминале, занимают терминальную сессию, и оболочка не выводит пользователю строку приглашения до тех пор, пока программа не завершится.

```
$ firefox csc.sibsutis.ru
```

Работу некоторых запущенных в терминале программ можно прервать с помощью сочетания клавиш «**Ctrl + c**» в окне терминала. В этот момент программе посылается сигнал **INT** (Interrupt).

Чтобы запустить программу в фоновом режиме необходимо завершить команду символом амперсанд «**&**». После этого в терминал выводится информация о запущенном процессе включая номер *задания* терминала, и приглашения пользователю на ввод новой команды.

```
$ top &
```

Используя команду **jobs** мы можем получить список заданий которые запущены через терминал.

```
$ jobs
```

Чтобы вернуть запущенный в фоне процесс на передний план используется команда **fg** с указанием номера задания из списка заданий.

```
$ fg %2
```

Если мы хотим перевести процесс в состояние *остановленный*, используется сочетание клавиш «**Ctrl + z**». В этот момент программе посылается сигнал TSTP (Terminal Stop).

После этого мы можем либо переместить задание на передний план командой **fg**, либо продолжить его выполнение в фоновом режиме командой **bg**.

```
$ bg %2
```

Еще одним способом управлять выполнением процессов является использование утилиты **kill**. Данная команды позволяет послать определенный сигнал процессу. Возможно завершение процесса как по имени, так и по номеру задания или по идентификатору PID процесса.

\$kill -SIGINT 124672

Получить список сигналов можно с помощью опции **-l**.

\$ kill -l

Послать сигнал нескольким процессам можно с помощью команды **killall**.

\$ killall gedit

Приведенная выше команда завершит все процессы текущего пользователя с именем **gedit**. По умолчанию команда отправляет сигнал **TERM** (software termination signal).

Таблица 2. Примеры команд

Команда	Описание
ps	Выдать снимок имеющихся в системе процессов.
top	Выдать информацию о процессах системы в интерактивной форме.
pstree	Вывод дерева процессов в системе
jobs	Выдать список заданий в текущей сессии оболочки.
fg	Переместить задание из текущей сессии на передний план.
bg	Переместить задание из текущей сессии в фон.
kill	Завершить процесс, отправить сигнал процессу.
killall	Завершить группу процессов, отправить сигнал группе процессов.

3 Порядок выполнения лабораторной работы

1. Прочитайте теоретический материал по лабораторной работе.
2. Ознакомьтесь с работой команд, приведенных в Таблице 2. Изучите для содержащихся в Таблице 2 команд страницы справочного руководства.
3. Создайте файл **proc1**, содержащий список процессов пользователя **root**, отсортированный по идентификатору родительского процесса. Используйте команду **ps** и изученные ранее утилиты.

4. Получите информацию о процессах вашего пользователя, имеющих статус *работоспособный*.
5. Добавьте к файлу **proc1** сведения о процессе в данный момент потребляющий большой процесс ресурсов центрального процессора.
6. Запустите утилиту **top**. Изучите содержимое информационных полей, предоставляемых утилитой. Получите информацию о степени использования ресурсов системы, количестве пользователей, времени работы системы.
7. Ознакомьтесь со справочной информацией утилиты **top**. Изучите действия, выполняемые по нажатию на клавиши, «**u**», «**f**», «**R**», «**d**», «**L**». Настройте вывод утилиты **top** в соответствии с заданием 3, с периодом обновления 1с.
8. В окне терминала запустите программу **firefox**. Используя команду **pstree** получите информацию о дереве процессов и их идентификаторах. Найдите поддерево для процесса **firefox**, изучите список составляющих его процессов.
9. В новом окне терминала запустите программу **firefox**. Затем два раза запустите утилиту **gedit** фоновом режиме.
10. Получите список заданий текущей сессии терминала.
11. Используя команды **fg** и **bg** и сочетания клавиш «**Ctrl + z**» и «**Ctrl + c**» научитесь перемещать задачи из фона на передний план и наоборот.
12. Получите список сигналов для команды **kill**. Завершите запущенные процессы с помощью команды **kill** с помощью сигналов **SIGKILL** и **SIGTERM**.
13. Выполните команду **killall -u имя_текущего_пользователя**.

3. Контрольные вопросы

1. Какие способы получения информации о процессах в системе вы знаете?
2. Как можно управлять выводом утилиты **top**?
3. Какие сигналы отправляются сочетаниями клавиш «**Ctrl + z**» и «**Ctrl + c**»?
4. Какой процесс в системе запускается первым?
5. Какие типы процессов вы знаете?
6. Что такое дескриптор процесса?