

ПРИЛОЖЕНИЯ UNIX СИСТЕМ

Лекция №5

Ст. преподаватель
кафедры ПДСиМ
Квиткова Ирина Геннадьевна

Сигналы

- **Сигналы** - способ передачи информации от одного процесса к другому или от ядра ОС к любому процессу.
- Сигналы напоминают программные прерывания – средства, с помощью которых нормальное выполнение процесса может быть прервано.
- Сигналы могут быть *синхронными* (инициирует сам процесс) и *асинхронными* (инициирует пользователь или ядро при контроле ошибок аппаратуры).
- Каждый сигнал имеет уникальное имя.
- Сигналы хранятся в виде мнемонической формы записи в файле <signal.h>.

Имя сигнала	Код	Комментарий
SIGHUP	1	Закончить процесс. При отключении связи с терминалом рассылается всем процессам.
SIGINT	2	Закончить процесс. Посылается всем процессам при нажатии определённых клавиш <Delete>, <Ctrl-C>.
SIGQUIT	3	Посылается при нажатии <Ctrl+\>.
SIGILL	4	Посылается текущему PID при выполнении недопустимых операций.
SIGTRAP	5	Ловушка отладчика.
SIGIOT	6	Текущему PID при неисправности в аппаратуре.

Имя сигнала	Код	Комментарий
SIGKILL	9	Адресуемому PID при выдаче системного вызова kill.
SIGBUS	7	Текущему PID при ошибке в косвенной адресации.
SIGSEVPG	11	Текущему PID при выходе за пределы сегмента.
SIGPIPE	13	Текущему PID при записи в pipe, когда нет читающего процесса.
SIGALARM	14	При срабатывании таймера.
SIGTERM	15	К адресуемому PID при выполнении команды kill в командном режиме.
SIGUSR1	10	Определяются пользователем.
SIGUSR2	12	
SIGSTOP	19	Остановка процесса по клавишам <Ctrl-Z>.

Процесс может выбрать одно из трех возможных действий при получении сигнала:

- *игнорировать сигнал;*
- *перехватить и самостоятельно обрабатывать его;*
- *позволить действие по умолчанию.*

Действия определяются функцией-обработчиком, созданной пользователем, или одним из следующих значений:

SIG-DFL - указывает ядру, что при получении процессом сигнала необходимо вызвать системный обработчик и выполнить действия по умолчанию.

SIG-IGN - сигнал следует игнорировать.

Для перехвата сигналов применяется команда

trap command condition

выполнить заданную команду **command** при получении определённого сигнала или события в системе (**condition**).

condition - символическое или числовое обозначение сигнала; **command** выполняется при получении этого сигнала.

Возможны ситуации:

- 1) **command** отсутствует или равен «-» - восстанавливается стандартное действие сигнала.
- 2) **command** - пустая строка – сигналы игнорируются.
- 3) **condition** =0 или **exit** – команды выполняются при завершении работы интерпретатора.

Например:

trap " " **1 2** - игнорирует *SIGHUP, SIGINT*.

trap **1 2 3** - восстанавливает действие *SIGHUP, SIGINT, SIGQUIT*.

➤ **trap -l** – отобразить полный список сигналов.

Средством послылки сигнала служит команда

kill -N^osign PID

Пример: послать с терминала сигнал завершения процессу, который запущен в фоновом режиме:

**./long-program &
kill \$!**

(\$! содержит PID последнего процесса, запущенного в фоне)

По умолчанию **kill** посылает **SIGTERM**.

Иногда процесс продолжает существовать при получении SIGTERM.

Тогда к нему применяется более жёсткое средство – сигнал **SIGKILL**, который нельзя ни перехватить, ни игнорировать:

kill -9 PID или

kill -SIGKILL PID

Уничтожить все процессы за исключением начального процесса-интерпретатора:

kill 0

Уничтожить все процессы с именем **proc**:

killall proc

Пример:

killall -SIGKILL top

принудительно завершить процесс top.

Именованный канал (FIFO)

FIFO - однонаправленное средство передачи данных, чтение которых происходит в порядке записи.

Это отдельный тип файла в ФС, обозначается буквой **p**rwх...

Чтобы создать FIFO, необходим системный вызов **mknod имя устройство** (или **mkfifo имя**)

Пример: 1. Создать программный канал с консоли:

```
mknod fifo p
```

2. Записать в программный канал что-нибудь:

```
cat >fifo
```

```
12345
```

3. Считывать содержимое с FIFO с другой консоли:

```
cat fifo
```

Сообщения

Чтобы послать сообщение другому пользователю, используется команда:

```
write login_name console  
text  
<Ctrl-D>
```

Например:

```
write A62_1 tty3
```

mesg y - *разрешить сообщения;*

mesg n – *запретить.*

Команды управления процессами

ps – вывод информации о запущенных процессах.

Ключи: **a** - процессы, связанные с текущим терминалом, и процессы других пользователей;

x - процессы, отсоединённые от терминала (демоны, службы);

aux - вывод всех процессов в системе;

-eH - вывод процессов в виде дерева;

-u user - процессы пользователя user;

-o format – вывод информации в определённом формате.

Пример:

ps -eo pid,cmd – вывести PID и имена всех процессов системы.

nohup команда

команда продолжает выполняться после выхода из системы. Результат выполнения команды сохранится в файле **nohup.out**.

nice -n value команда

изменение приоритета запускаемого процесса *команда* на значение, равное *value* (может быть от -20 до 19, в порядке уменьшения приоритета).

renice -value PID

изменение приоритета запущенного процесса с PID на значение, равное *value*.

Пример:

renice +5 1248 – понижение приоритета процесса с номером 1248 на 5.

Чтобы запустить процесс в заданное время, используется команда:

**at время
любые команды...
ctl-d**

или из файла:

at 3am << файл

jobs - список остановленных и фоновых задач.

bg number - продолжить выполнение остановленной задачи в фоне, имеющей номер *number* (указан в выводе **jobs**).

fg number – перевести задачу с номером *number* из фонового режима в активный.

top

показать все запущенные процессы в интерактивном режиме.

Клавиши для управления командой:

h - справка о программе;

k - уничтожить процесс;

n - число отображаемых процессов;

u - сортировать по имени пользователя;

M - сортировать по объему ОЗУ;

P - сортировать по загрузке ЦП;

r - изменить приоритет выполнения;

q – ВЫХОД.

ps tree - отобразить дерево процессов:

-p – показывает PID процессов;

-u – показывает имя user, запустившего процесс.