Факультет
биоинженерии и биоинформатики
Московский государственный университет имени М.В.Ломоносова

Управление выполнением программ

Процесс

```
# TIME_STYLE='+%R' ls -l /proc/22130
dr-xr-xr-x  2 root root 0 21:08 attr
-rw-r--r--  1 root root 0 21:08 autogroup
-r--------  1 root root 0 21:08 auxv
-r--r--r--  1 root root 0 21:08 cgroup
--w-------  1 root root 0 21:08 clear_refs
-r--r--r--  1 root root 0 21:08 cmdline
-rw-r--r--  1 root root 0 21:08 comm
-rw-r--r--  1 root root 0 21:08 coredump_filter
-r--r--r--  1 root root 0 21:08 cpuset
lrwxrwxrwx  1 root root 0 21:08 cwd -> /
-r--------  1 root root 0 21:08 environ
lrwxrwxrwx  1 root root 0 21:08 exe -> /bin/bash
dr-x------  2 root root 0 21:08 fd
dr-x------  2 root root 0 21:08 fdinfo
-rw-r--r--  1 root root 0 21:08 gid_map
-r--------  1 root root 0 21:08 io
-r--r--r--  1 root root 0 21:08 limits
-rw-r--r--  1 root root 0 21:08 loginuid
dr-x------  2 root root 0 21:08 map_files
-r--r--r--  1 root root 0 21:08 maps
-rw-------  1 root root 0 21:08 mem
-r--r--r--  1 root root 0 21:08 mountinfo
-r--r--r--  1 root root 0 21:08 mounts
-r--------  1 root root 0 21:08 mountstats
dr-xr-xr-x 59 root root 0 21:08 net
dr-x--x--x  2 root root 0 21:08 ns
-r--r--r--  1 root root 0 21:08 numa_maps
-rw-r--r--  1 root root 0 21:08 oom_adj
-r--r--r--  1 root root 0 21:08 oom_score
-rw-r--r--  1 root root 0 21:08 oom_score_adj
-r--------  1 root root 0 21:08 pagemap
-r--------  1 root root 0 21:08 patch_state
-r--------  1 root root 0 21:08 personality
-rw-r--r--  1 root root 0 21:08 projid_map
lrwxrwxrwx  1 root root 0 21:08 root -> /
-rw-r--r--  1 root root 0 21:08 sched
-r--r--r--  1 root root 0 21:08 schedstat
-r--r--r--  1 root root 0 21:08 sessionid
-rw-r--r--  1 root root 0 21:08 setgroups
-r--r--r--  1 root root 0 21:08 smaps
-r--r--r--  1 root root 0 21:08 smaps_rollup
-r--------  1 root root 0 21:08 stack
-r--r--r--  1 root root 0 21:08 stat
-r--r--r--  1 root root 0 21:08 statm
-r--r--r--  1 root root 0 21:08 status
-r--------  1 root root 0 21:08 syscall
dr-xr-xr-x  3 root root 0 21:08 task
-r--r--r--  1 root root 0 21:08 timers
-rw-rw-rw-  1 root root 0 21:08 timerslack_ns
-rw-r--r--  1 root root 0 21:08 uid_map
-r--r--r--  1 root root 0 21:08 wchan
```

Группа процессов

Процессы в рамках одной задачи (job) пользовательской сессии.

- процессам группы присвоен одинаковый PGID (или PGRP)
- процесс с PID = PGID называется лидером группы
- имеет смысл управлять всеми процессами группы одновременно

Встроенная команда bash `kill` позволяет посылать сигналы целой группе процессов (программа kill этого не умеет).

Сессии

Терминальная сессия – процессы с общим управляющим терминалом (или псевдотерминалом), имеют общий SID.

Пользовательская сессия – все процессы, порожденные после входа пользователя в систему.

Пользовательские сессии контролируются менеджером systemd-logind, ими можно управлять с помощью `loginctl` .

Ограничение ресурсов

Приоритет процесса

Niceness – (инвертированный) приоритет процесса с точки зрения планировщика.

- меняется в интервале от -20 (самый высокий приоритет) до 19 (самый низкий)
- по умолчанию программы запускаются с приоритетом 0
- можно задать приоритет при запуске с помощью `nice`
- или изменить в процессе работы с помощью `renice` или `top`
- пользователи обычно могут только снижать приоритет своих процессов

ulimit

```
$ cat /proc/60949/limits
Limit                     Soft Limit           Hard Limit           Units
Max cpu time              unlimited            unlimited            seconds
Max file size             unlimited            unlimited            bytes
Max data size             unlimited            unlimited            bytes
Max stack size            8388608              unlimited            bytes
Max core file size        0                    unlimited            bytes
Max resident set          unlimited            unlimited            bytes
Max processes             78117                78117                processes
Max open files            1024                 1048576              files
Max locked memory         2567227392           2567227392           bytes
Max address space         unlimited            unlimited            bytes
Max file locks            unlimited            unlimited            locks
Max pending signals       78117                78117                signals
Max msgqueue size         819200               819200               bytes
Max nice priority         0                    0
Max realtime priority     0                    0
Max realtime timeout      unlimited            unlimited            us
```

Встроенная команда bash `ulimit` позволяет менять некоторые ограничения.

cgroups

Новый механизм ядра Linux для ограничения, учета и изоляции ресурсов.

- иерархические группы процессов с установленным ограничением ресурса
- взаимодействие через виртуальную ФС cgroup2 (/sys/fs/cgroup)
- systemd позволяет ограничивать ресурсы для юнитов (системных сервисов, пользовательских сессий и т.д.)

systemd-run

Позволяет обычным пользователям создавать временный юниты для запуска программ, в том числе, с возможностью ограничения ресурсов.

Запустить a-program с ограниченными ресурсами:

```
$ systemd-run --scope --user -p 'MemmoryHigh=1G' -p 'CPUQuota=200%' a-program
```

Подробно ограничение ресурсов для юнитов описано в `man systemd.resource-control` .

Мониторинг исполнения

Потребление системных ресурсов

При запуске вычислений следует контролировать:

- загрузку процессора
- количество потоков исполнения
- потребление оперативной памяти
- потребление swap
- свободное место на разделах диска
- размеры создаваемых файлов

top

Выводит отсортированный обновляющийся список процессов в интерактивном режиме.

```
top - 19:06:47 up 415 days,  3:25,  1 user,  load average: 0.17, 0.12, 0.06
Threads: 772 total,   1 running, 771 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.5 us,  0.1 sy,  0.0 ni, 99.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  64298.2 total,   2354.4 free,   6073.9 used,  55869.9 buff/cache
MiB Swap:  30518.0 total,  30017.5 free,    500.5 used.  57535.6 avail Mem

    PID    PPID nTH    TGID USER      NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
3136433       1   1 3136433 opendkim   0   23116    960    648 S   0.0   0.0   0:00.00 /sbin/opendkim
3136434 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   2:20.41 /sbin/opendkim
3136435 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   0:00.00 /sbin/opendkim
3136436 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   0:00.00 /sbin/opendkim
3136437 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   2:02.18 /sbin/opendkim
3138090 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   1:31.11 /sbin/opendkim
3704485 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   3:12.84 /sbin/opendkim
 285358 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   0:23.19 /sbin/opendkim
```

ps

Неинтерактивно выводит запрошенную информацию о выбранных процессах.

```
$ ps -u root --sort='-%mem' -o 'uid,pid,ppid,ni,tty,cmd' | head
  UID     PID    PPID  NI TT       CMD
    0     734     719   0 tty7     /usr/lib/xorg/Xorg :0 -seat seat0
    0     278       1   0 ?        /lib/systemd/systemd-journald
    0     585       1   0 ?        /usr/libexec/udisks2/udisksd
    0    1068     719   0 ?        lightdm --session-child 12 21
    0     640       1   0 ?        /usr/sbin/ModemManager
    0       1       0   0 ?        /sbin/init
    0     559       1   0 ?        /usr/libexec/polkitd --no-debug
    0     304       1   0 ?        /lib/systemd/systemd-udevd
    0    1262       1   0 ?        /usr/libexec/upowerd
$ ps $(pgrep bash)
    PID TTY      STAT   TIME COMMAND
  22800 pts/1    Ss     0:00 bash
  22806 pts/2    Ss     0:02 bash
$
```

Сигналы

Приостановка и прерывание исполнения

`SIGINT (2)` – сигнал прерывания с клавиатуры (`Ctrl+C`)
`SIGQUIT (3)` – сигнал прерывания с сохранением дампа памяти (`Ctrl+\`)
`SIGKILL (9)` – сигнал немедленного прерывания (убийства)
`SIGTERM (15)` – сигнал прерывания
`SIGCONT (18)` – сигнал продолжения выполнения после приостановки
`SIGSTOP (19)` – сигнал приостановки
`SIGTSTP (20)` – сигнал приостановки с клавиатуры (`Ctrl+Z`)

Сигналы, связанные с вводом/выводом

`SIGHUP (1)` – сигнал об обрыве связи с управляющим терминалом
`SIGPIPE (13)` – broken pipe
`SIGTTIN (21)` – попытка чтения из терминала фоновым процессом
`SIGTTOU (22)` – попытка записи в терминал фоновым процессом

Обработка сигналов

Для обработки сигналов в bash есть встроенная команда trap

```
Игнорировать стандартный сигнал прерывания
$ trap '' SIGTERM

Изменить обработчик SIGINT
$ trap 'echo -ne \\nПривет' INT # можно без SIG
$ ^C
Привет
$ ^C
Привет
$

Вернуть стандартный обработчик SIGINT
$ trap 2 # можно указать номер сигнала
$ ^C
$
```