



Российский разработчик и поставщик
решений на основе программного обеспечения
с открытым исходным кодом

Понятие о процессах в РЕД ОС

Базовые определения

Процесс (process) — блок адресного пространства в котором выполняются экземпляр программы. Процесс может запускать другие процессы

Дескриптор файла — некоторое число, которое используется для обращения к файлу. При запуске процесс наследует дескрипторы от родительского процесса.

Идентификатор родительского процесса (parent process ID) указывает на родительский процесс.

Идентификатор группы процессов (process group ID). Процессы могут объединяться в группы. Каждая группа обозначается идентификатором группы. Процесс, идентификатор которого совпадает с идентификатором группы, называется лидером группы.

Идентификатор сеанса (session ID). Каждая группа процессов принадлежит к сеансу. Сеанс связывает процессы с управляющим терминалом. Когда пользователь входит в систему, все создаваемые им процессы будут принадлежать сеансу, связанному с его текущим терминалом.

Базовые определения

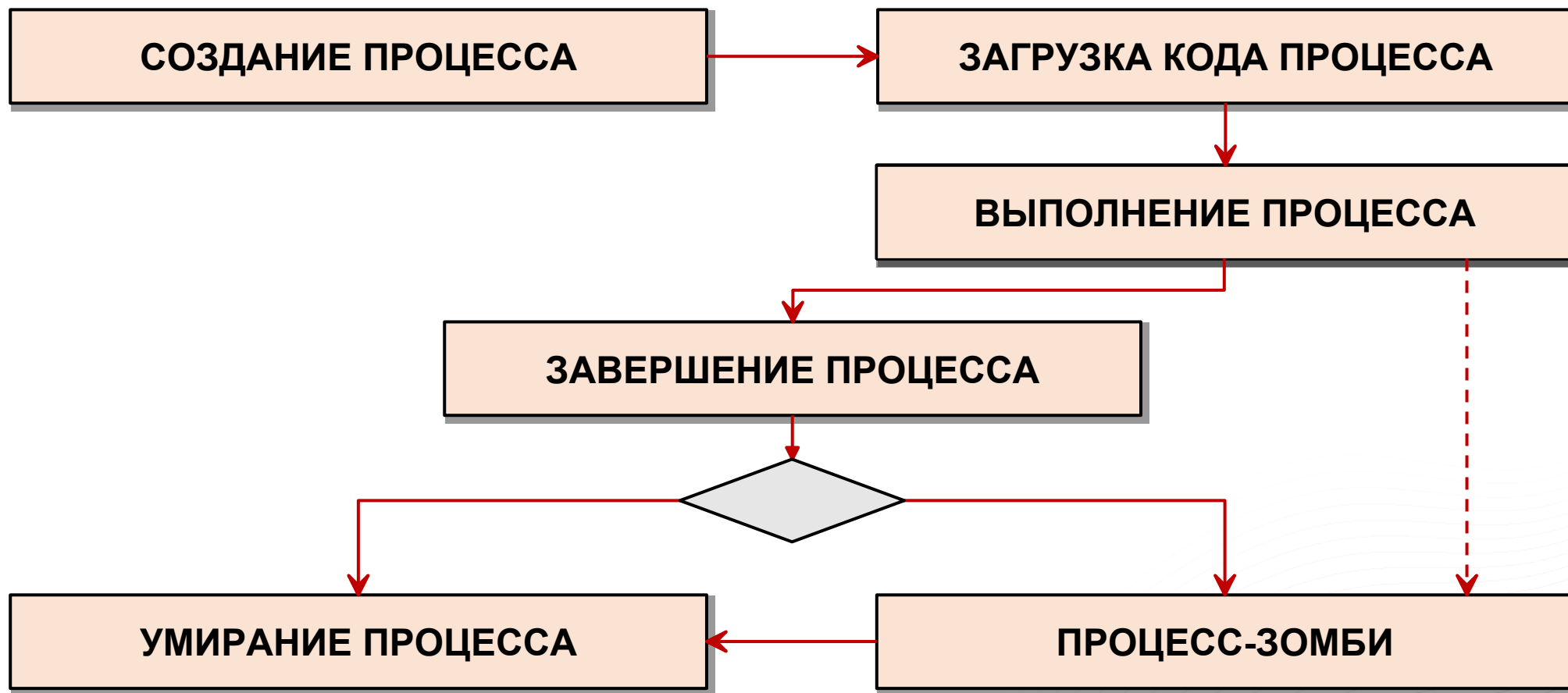
Программное окружение (programm environment) это набор строк, заканчивающихся нулевым символом. Строки называются переменными окружения и имеют следующий формат: имя переменной = значение переменной. Параметры, необходимые для программы – **окружение процесса**.

Текущий рабочий каталог это каталог от которого происходит разрешение относительных имен.

Текущий корневой каталог это каталог от которого производится разрешение абсолютных имен. Выше корневого каталога файлы не доступны.

Приоритет (nice). Значение nice ("дружелюбность") показывает готовность процесса уступить свое процессорное время другим процессам. Чем больше значение nice, тем ниже приоритет процесса.

Базовые определения



Идентификатор процесса

Идентификатор процесса (process ID) это целое число однозначно идентифицирующее процесс. Процесс с идентификатором 1 это процесс init.

Идентификаторы пользователя и группы. С процессами связаны так же:

- реальный идентификаторы пользователя и группы (real userID, real groupID), указывающие кто создал процесс.
- эффективные идентификаторы пользователя и группы (effective userID, effective groupID), определяющие права процесса в системе.

Идентификаторы процессов (PID и PPID) лежат в диапазоне до

`cat /proc/sys/kernel/pid_max`

Процессы, не привязанные к конкретному терминалу – **«демоны»(daemon)**.

Подробная информация о процессе

Для отображения состояния процессов существуют виртуальные папки в корневом каталоге Linux:

- **/proc** — файловая система **proc** действует как интерфейс к внутренним структурам данных **о** запущенных **процессах в ядре**. В **она** также может использоваться для получения информации о ядре **и** изменения определенных **параметров ядра** во время выполнения (sysctl).
- **/sys** — файловая система **sysfs** представляет средство для экспорта структур **данных ядра, их атрибутов** и связей между ними в пространство пользователя. В **/sys** дублируется информация, не относящаяся к процессу, которая попала в дерево **/proc**.

Подробная информация о процессе

cat /proc/<PID>/status — подробный вывод статуса

cat /proc/<PID>/syscall — Адрес процесса в ячейках оперативной памяти

cat /proc/<PID>/cmdline — Команда, которой был запущен процесс

ll /proc/<PID>/cwd — символьная ссылка на рабочий каталог процесса

ll /proc/<PID>/exe — Символьная ссылка на исполняемый файл процесса

ll /proc/<PID>/fd/ — Увидеть ссылки на дескрипторы открытых файлов, которые затрагивает процесс

Межпроцессное взаимодействие

Виды механизмов межпроцессного взаимодействия:

Обмен данными

- Каналы и очереди FIFO
- Сокеты (локальные и сетевые)
- Очереди сообщений
- Разделяемая память

Синхронизация работы процессов

- Семафоры
- Блокировки файлов

Сигналы

Состояние процесса

Текущее состояние процесса. Может принимать разные значения:

R — выполнимый процесс;

S — спящий;

D — в состоянии подкачки на диске;

T — остановлен;

Z — зомби;

W — не имеет резидентных страниц;

< — высоко-приоритетный;

N — низко-приоритетный;

L — имеет страницы, заблокированные в памяти.

I — бездействующий поток ядра;

s — руководитель сессии.

Команды для работы с процессами

Команда **ps** принимает несколько различных типов параметров:

- Параметры стиля UNIX, которым предшествует тире.
- Параметры стиля BSD, используемые без тире.
- Параметры GNU с двумя тире перед ними

В некоторых случаях могут возникать конфликты ключей, поэтому лучше придерживаться одного типа опций.

Команды для работы с процессами

Форма BSD утилиты **ps**

ps aux

f - отображает древо процессов

--sort=-%mem - можно сортировать, например по использованию памяти

o - отображает пользовательские столбцы, например pid,comm

ps axo pid,comm,%mem --sort=%mem

```
[dima@host1 proc]$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.2	241380	12412	?	Ss	17:19	0:02	/usr/lib/systemd/systemd
root	2	0.0	0.0	0	0	?	S	17:19	0:00	[kthreadd]
...										

Команды для работы с процессами

ps [-axewjlu] [-o формат] [-U пользователь] [-p pid]

```
[july@localhost ~]$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	224820	17364	?	Ss	09:55	0:02	/usr/lib/syst
root	2	0.0	0.0	0	0	?	S	09:55	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	09:55	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	09:55	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	09:55	0:00	[slub_flushw

VSZ — Virtual Set Size. Виртуальный размер процесса (в килобайтах).

RSS — Resident Set Size. Размер резидентного набора (количество 1К-страниц в памяти).

TTY — терминал, из под которого был запущен процесс.

STAT — текущее состояние процесса.

START — дата запуска процесса.

TIME — время потраченное процессором на процесс.

COMMAND — команда, запустившая процесс.

Команды для работы с процессами

Форма UNIX утилиты **ps**

ps [-axewjlu] [-o формат] [-U пользователь] [-p pid]

Значения параметров следующие:

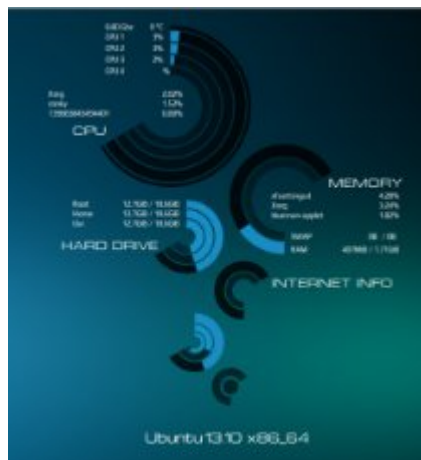
- a** вывести информацию о процессах всех пользователей.
- x** вывести информацию о процессах не подключенных к терминалу.
- e** вывести значения переменных окружения процесса.
- w** использовать строки длиной 132 символа. Если указан несколько раз, то строки не обрезаются совсем.
- j**, -**l**, -**u** - меняют формат вывода информации.
- o** формат вывести информацию в указанном формате.
- U** пользователь вывести информацию о процессах указанного пользователя.
- p pid** вывести информацию о процессе с указанным идентификатором.

Команды для работы с процессами

Conky — это настраиваемый системный монитор-виджет.

Конфигурация **/etc/conky/conky.conf**

Программа настройки **conky-manager**



48-generic on x8

Время работы: 0h 56m 14s

Frequency (in MHz): 2037

Frequency (in GHz): 2,04

RAM Usage: 6,85GiB/15,6GiB - 43%

Swap Usage: 0B /0B - 0%

CPU Usage: 1%

Processes: 515 Running: 0

File systems:

/ 30,5GiB/36,4GiB

/home 83,9GiB/143GiB

Networking:

Up: 0B - Down: 0B

Name	PID	CPU%	MEM%
chrome	10624	0,34	2,44
systemd	1	0,17	0,08
containerd	1104	0,17	0,30
Xorg	6117	0,17	0,34

Команды для работы с процессами

Утилита top

Опций запуска у команды не много и использовать их активно не принято, потому что большинство действий выполняются с помощью интерактивных команд.

Программа имеет четыре окна для вывода данных. Это **def**, **job**, **mem** и **usr**. Каждое окно выделяется другим цветом в цветном режиме, а также содержит разный набор колонок. Для просмотра всех окон используйте команду **A**, а для переключения между ними - **a**

```
1:Def - 21:37:12 up 12:47, 0 users, load average: 1,91, 2,14, 2,12
Tasks: 324 total, 2 running, 320 sleeping, 0 stopped, 2 zombie
%Cpu(s): 8,8 us, 5,4 sy, 0,0 ni, 85,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 5791,4 total, 254,6 free, 3454,3 used, 2082,4 buff/cache
MiB Swap: 0,0 total, 0,0 free, 0,0 used. 1784,4 avail Mem
```

1	PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
	4982	dima	20	0	1125,3g	503192	121128	R	45,4	8,5	50:15.45	yandex_browser
	4805	dima	20	0	33,1g	285216	144664	S	17,4	4,8	257:19.03	yandex_browser
	3144	root	20	0	1495652	172616	104832	S	13,8	2,9	162:54.41	Xorg

2	PID	PPID	TIME+	%CPU	%MEM	PR	NI	S	VIRT	RES	UID	COMMAND
	133860	3110	0:00.01	0,3	0,0	20	0	Z	0	0	0	loginctl
	133661	3209	0:00.01	0,0	0,1	20	0	S	61020	8060	0	systemd-userwor
	133588	96733	0:01.16	1,3	0,1	20	0	R	233064	4812	1000	top
	133565	3209	0:00.01	0,0	0,1	20	0	S	61020	8060	0	systemd-userwor

3	PID	%MEM	VIRT	RES	CODE	DATA	SHR	nMaj	nDRT	%CPU	COMMAND
	4645	9,3	2765628	551016	98860	2008508	146328	2794	0	0,0	Telegram
	4982	8,5	1125,3g	503192	164124	854460	121128	283	0	45,4	yandex_browser
	117956	7,3	3003080	432488	4	423708	200100	1515	0	0,3	soffice.bin
	55421	6,8	1129,3g	405680	164124	740344	124764	36	0	2,0	yandex_browser

4	PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
	711	1	192	systemd+	systemd+	?	0:00.45	0,0	0,2	S	systemd-network
	2558	1	172	rtkit	rtkit	?	0:00.51	0,0	0,1	S	rtkit-daemon
	3101	1	29	rpcuser	rpcuser	?	0:00.02	0,0	0,9	S	rpc.statd
	2496	1	32	rpc	rpc	?	0:00.10	0,0	0,1	S	rpcbind

Команды для работы с процессами

Утилита **top**. Некоторые клавиши интерактивных команд управления.

1 — скрыть/показать верхнюю строку

t — переключение вида отображения использования процессора

m — переключение вида использования памяти

b — отключить/включить жирное выделение важных процессов

z — включить/отключить цветное выделение

x — включить/отключить выделение главного столбца

c — включить/отключить показ команд запуска

u — включить выборку по пользователю (! - исключить пользователя)

v — включить/отключить дерево процессов

i — включить/отключить только активные процессы

s — задать интервал отображения информации

Shift+< или shift+> — изменение активного столбца сортировки

Команды для работы с процессами

Утилита top

Для использования top в скриптах или для grep необходимо включить пакетный режим и указать количество повторений опроса параметров

top -b -n 1 | grep chromium

После всех манипуляций можно сохранить настройки программы клавишей **W**. Запишется конфигурационный файл, top запустится в следующий раз с теми же настройками.

Для сброса конфигурации можно удалить файл настроек.

Обычно это файл **~/.config/procps/toprc**.

Команды для работы с процессами

Утилита htop. Для изменения внешнего вида — панели с информацией о системе, выводимых столбцах и прочем, нажмите кнопку **F2** или **S (Shift+s)**.

```
1  [|||||] 15.7% 5  [||||] 9.7%
2  [||||] 8.6% 6  [||||] 7.9%
3  [||||] 11.0% 7  [||||] 10.5%
4  [||||] 13.9% 8  [||||] 13.6%
Mem[|||||] 6.41G/31.3G Tasks: 183, 962 thr; 1 running
Swp[ ] 0K/15.7G Load average: 0.92 1.03 1.24
Uptime: 03:33:47
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
16129	july	20	0	1.1T	163M	105M	S	17.6	0.5	4:49.38	/opt/yandex/browser/yandex_browse
1597	root	20	0	1434M	288M	232M	S	14.4	0.9	22:31.02	/usr/libexec/Xorg :0 -background
2335	july	20	0	1356M	63380	43068	S	9.8	0.2	3:36.27	mate-panel
2082	july	9	-11	1527M	19252	13804	S	8.5	0.1	13:15.69	/usr/bin/pulseaudio --daemonize=n
7904	july	20	0	1.1T	348M	126M	S	5.2	1.1	45:47.38	/opt/yandex/browser/yandex_browse
4688	july	20	0	4805M	331M	181M	S	5.2	1.0	8:12.44	/app/extra/viber/Viber
17574	july	20	0	2238M	556M	180M	S	4.6	1.7	4:39.98	/usr/lib64/libreoffice/program/so
16135	july	20	0	1.1T	163M	105M	S	3.9	0.5	1:11.88	/opt/yandex/browser/yandex_browse
29911	july	20	0	217M	5236	3328	R	2.6	0.0	0:02.63	htop
3108	july	20	0	33.5G	230M	137M	S	2.6	0.7	57:07.98	/opt/yandex/browser/yandex_browse
4839	july	20	0	4805M	331M	181M	S	2.6	1.0	4:25.72	/app/extra/viber/Viber
2977	july	20	0	40.9G	232M	129M	S	2.6	0.7	4:35.55	/usr/share/skypeforlinux/skypefor
7915	july	20	0	1.1T	348M	126M	S	2.6	1.1	8:47.26	/opt/yandex/browser/yandex_browse

```
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit
```

Команды для работы с процессами

Утилита **atop** - Расширенный системный и технологический монитор. Собирает статистику с помощью сервиса. Конфиг **/etc/sysconfig/atop**. Лог **/var/log/atop/***

Наиболее часто используемые интерактивные команды:

d — сортировка по использованию диска;
m — сортировка по занятой памяти;
v — подробно о процессах;
u — сортировка по пользователям;
i — изменение времени проверки;
g — вернет все в дефолтный вывод;
c — полный путь к файлу процесса
и др.

Просмотр логов

atop -r /var/log/atop/atop_20201009

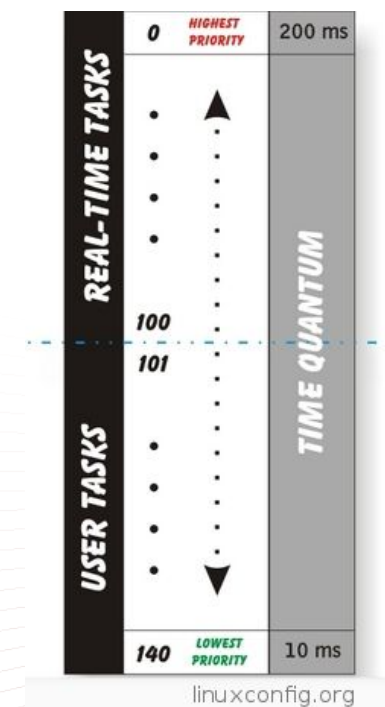
Навигации по времени в логе:

t - вперед по времени; **Shift + t** - назад по времени.

ATOP - localhost										2023/06/29 13:29:29		-----		3h34m29s elapsed			
PRC	sys	51m47s	user	3h24m	#proc	334	#tslpu	92	#zombie	1	#exit	1					
CPU	sys	26%	user	96%	irq	7%	idle	650%	wait	21%	curscal	32%					
CPL	avg1	1.20	avg5	1.10	avg15	1.26	csw	224604e3	intr	12826e4	numcpu	8					
MEM	tot	31.3G	free	21.8G	cache	3.9G	buff	216.1M	slab	346.0M	numnode	1					
SWP	tot	15.7G	free	15.7G	swcac	0.0M			vmcom	17.2G	vmlim	31.3G					
PSI	cpusome	1%	memsome	0%	memfull	0%	iosome	3%	iofull	3%	cs	2/1/1					
LVM	o_redos-home	busy	1%	read	22708		write	220891	MBw/s	0.2	avio	0.54 ms					
LVM	o_redos-root	busy	0%	read	48216		write	37333	MBw/s	0.0	avio	0.61 ms					
LVM	o_redos-swap	busy	0%	read	99		write	0	MBw/s	0.0	avio	0.71 ms					
DSK	sdb	busy	1%	read	51497		write	138669	MBw/s	0.3	avio	0.95 ms					
DSK	sda	busy	0%	read	1053		write	6	MBw/s	0.0	avio	9.51 ms					
NET	transport		tcpi	233944	tcpo	166806	udpi	404259	udpo	70685	tcpao	1909					
NET	network		ipi	627371	ipo	226138	ipfrw	0	deliv	623458	icmpo	55					
NET	wlp3s0	0%	pcki	147938	pcko	994	sp	122 Mbps	si	106 Kbps	so	0 Kbps					
NET	eno1	0%	pcki	476247	pcko	234393	sp	1000 Mbps	si	328 Kbps	so	39 Kbps					
Number of variable resources limited to fit in this window																	
	PID	SYS	CPU	USR	CPU	RDELAY	VGROW	RGROW	RUID	ST	EXC	THR	S	CPUNR	CPU	CMD	1/56
	3108	15m59s	41m11s	1m41s	33.5G	230.2M	july	N-	-	13	S	2	27%	yandex_browser			
	7904	4m07s	41m45s	88.73s	1.1T	349.5M	july	N-	-	25	S	0	22%	yandex_browser			
	3308	2m16s	23m34s	53.75s	1.1T	457.0M	july	N-	-	21	S	2	12%	yandex_browser			
	3830	58.30s	23m23s	29.73s	1.1T	357.6M	july	N-	-	19	S	2	11%	yandex_browser			
	1597	7m50s	14m47s	29.05s	1.4G	288.5M	root	N-	-	6	S	7	11%	Xorg			
	2082	6m45s	6m34s	54.23s	1.5G	18.8M	july	N-	-	3	S	1	6%	pulseaudio			

Управление приоритетом процесса

- Алгоритмы планирования разделения времени (по умолчанию) и реального времени
- У процессов реального времени приоритет статический от 1 до 99
- У процессов разделения времени: статический приоритет равный 0 и динамический приоритет
- Динамический приоритет x зависит от числа NICE (-20 +19). Только root может уменьшать NICE.
- Для установки NICE **нового** процесса — **nice -n число команда**
- Для изменения NICE **запущенного** процесса — **renice -n число -p PID**



Сигналы

Kill посылает **сигнал** процессу. Обычно используется для завершения процесса (прерывание процесса). Получить список всех сигналов в табличном виде:

kill -L

Название	Код	Действие по умолчанию	Описание	Тип
SIGINT	2	Завершение	Сигнал прерывания (Ctrl-C) с терминала	Управление
SIGQUIT	3	Завершение с дампом памяти	Сигнал «Quit» с терминала (Ctrl-I)	Управление
SIGABRT	6	Завершение с дампом памяти	Сигнал посылаемый функцией abort()	Управление
SIGKILL	9	Завершение	Безусловное завершение	Управление
SIGTERM	15	Завершение	Сигнал завершения (сигнал по умолчанию для утилиты kill)	Управление
SIGTSTP	20	Остановка процесса	Сигнал остановки с терминала (Ctrl-Z).	Управление
SIGSTOP	19	Остановка процесса	Остановка выполнения процесса	Управление
SIGCONT	18	Продолжить выполнение	Продолжить выполнение ранее остановленного процесса	Управление
SIGTTIN	21	Остановка процесса	Попытка чтения с терминала фоновым процессом	Управление
SIGTTOU	22	Остановка процесса	Попытка записи на терминал фоновым процессом	Управление

Сигналы

По умолчанию команда kill передает сигнал SIGTERM и имеет числовое значение 15.

kill опции PID1 PID2 PID3

Сигнал может задаваться **числом** или **названием**.

Узнать PID процесса можно разными способами, например:

ps -ef | grep firefox

```
[july@localhost ~]$ pgrep -l yandex  
3065 yandex_browser
```

Pstree

```
[july@localhost ~]$ pstree
systemd--ModemManager--2*[{ModemManager}]
      --NetworkManager--2*[{NetworkManager}]
      --accounts-daemon--2*[{accounts-daemon}]
      --alsactl
      --anydesk--anydesk--2*[{anydesk}]
      --atd
      --atop
      --atopacctd
      --auditd--sedispatch
                        --2*[{auditd}]
      --avahi-daemon--avahi-daemon
      --bwrap--xdg-dbus-proxy--{xdg-dbus-proxy}
      --chrome_crashpad--2*[{chrome_crashpad}]
```

Lsof

Lsof - Утилита, служащая для вывода информации о том, какие файлы используются теми или иными процессами

```
30874 pts/1      00:00:00 mc
30876 pts/2      00:00:00 bash
31034 pts/2      00:00:00 ps
[july@localhost ~]$ lsof -p 30874
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
mc	30874	july	cwd	DIR	253,2	4096	21757953	/home/july
mc	30874	july	rtd	DIR	253,0	4096	2	/
mc	30874	july	txt	REG	253,0	1026048	464435	/usr/bin/mc
mc	30874	july	mem	REG	253,0	80384	396393	/usr/lib64/libnss_files-2.28.so
mc	30874	july	mem	REG	253,0	150743	461423	/usr/share/locale/ru/LC_MESSAGES/gl
o								
mc	30874	july	mem	REG	253,0	217750496	399501	/usr/lib/locale/locale-archive
mc	30874	july	mem	REG	253,0	325992	396395	/usr/lib64/libpthread-2.28.so

Pidstat

Утилита **pidstat**

Сокращение от PID Statistics (статистика по PID). Выдает различные статистические данные по процессам

```
[dima@RedOS ~]$ pidstat
Linux 5.15.78-2.el7.3.x86_64 (RedOS)    29.01.2023    _x86_64_    (8 CPU)

20:31:47      UID      PID    %usr  %system  %guest    %wait    %CPU   CPU  Command
20:31:47        0        1    0,03   0,05    0,00    0,00   0,09    1  systemd
20:31:47        0        2    0,00   0,00    0,00    0,00   0,00    2  kthreadd
20:31:47        0        9    0,00   0,39    0,00    0,06   0,39    4  kworker/u32:0+even
```


Strace

Strace - это утилита для диагностики, отладки и обучения пользователей Linux. Он используется для мониторинга и вмешательства во взаимодействие между процессами и ядром Linux, включая системные вызовы, доставку сигналов и изменения состояния процесса.

Посмотреть всё что делает утилита cat

strace cat 1.txt

Посмотреть только действие открытия

strace -e trace=openat cat 1.txt

Управление заданиями

Задачу (команду или скрипт) в фоновом режиме: **pluma &**

Также можно нажать **CTRL+Z**, но процесс при этом встанет на паузу.

Продолжить её выполнение в фоне — команда **bg**.

Список фоновых задач текущего терминала **jobs** или **jobs -l**

Вернуть задачу из фона — **fg [номер]**

Прервать выполнение задачи в фоне — **kill %номер**

Отсоединить задачу от терминала — **disown %номер**

Практическая работа

1. Поясните значения столбцов команды `ps -efl | more`
2. В консоли суперпользователя запустите утилиту `top` для текущего контроля процессов.
3. Из первой консоли создайте процесс `# od /dev/zero > /dev/null`. (С помощью команды `top` найдите и идентифицируйте запущенный процесс, найдите по идентификатору PPID его «родителя», определите его приоритет, долю загрузки центрального процессора %CPU и оперативной памяти %MEM.
4. Запустите аналогичную команду обычным пользователем. Сравните показатели процессов у разных пользователей с разными правами
5. Удалите созданные процессы командой `kill`.



Спасибо за внимание!

www.red-soft.ru
redos@red-soft.ru

