



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № 4

Название: Исследование способов организации оперативной памяти и взаимодействия процессов

Дисциплина: Операционные системы

Студент

ИУ6-52Б

(Группа)

(Подпись, дата)

С.В. Астахов

(И.О. Фамилия)

Преподаватель

А.М. Суровов

(Подпись, дата)

(И.О. Фамилия)

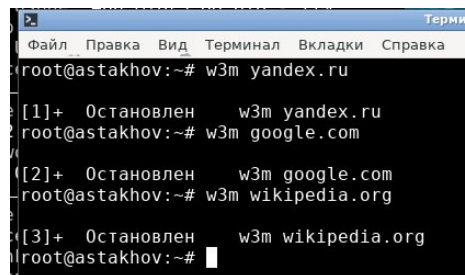
Москва, 2021

Цель работы: получение теоретических и практических сведений об управлении процессами, потоками и оперативной памятью в UNIX-подобных системах и в Linux в частности.

Задание:

1. В текстовом браузере некую страницу и перевести его в фоновый режим.
2. Запустить ещё два экземпляра текстового браузера в фоновом режиме.

Практическая часть: Откроем в w3m произвольную веб-страницу и переведем его в фоновый режим нажатием “Ctrl+Z”. Повторим эти действия еще 2 раза.



```
root@astakhov:~# w3m yandex.ru
[1]+  Остановлен   w3m yandex.ru
root@astakhov:~# w3m google.com
[2]+  Остановлен   w3m google.com
root@astakhov:~# w3m wikipedia.org
[3]+  Остановлен   w3m wikipedia.org
root@astakhov:~#
```

Рисунок 1 - перевод w3m в фоновый режим

Задание:

3. Найти процесс, максимально нагружающий процессор.

Практическая часть: воспользуемся командой “top”. Как видно, больше всего процессор нагружает conky.



PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1531	root	20	0	89224	13580	10956	S	1,7	0,3	0:45.02	conky
1434	root	19	-1	226064	60464	31844	S	0,3	1,5	0:19.48	Xorg
1622	root	20	0	115020	34560	27984	S	0,3	0,8	0:01.21	xfce4-terminal
1	root	20	0	36416	9128	7496	S	0,0	0,2	0:01.56	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp

Рисунок 2 - утилита “top”

Задание:

4. Вывести список процессов текущего пользователя
5. «Убить» первый процесс браузера в котором открыта 1 страница

Практическая часть: для просмотра процессов воспользуемся командой “ps” и убьем процесс с помощью “kill -9”.

```
Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
PID TTY      TIME CMD
1626 pts/0    00:00:00 bash
1630 pts/0    00:00:00 w3m
1635 pts/0    00:00:00 sh
1636 pts/0    00:00:00 w3mimgdisplay
1637 pts/0    00:00:00 w3m
1645 pts/0    00:00:00 sh
1646 pts/0    00:00:00 w3mimgdisplay
1647 pts/0    00:00:00 w3m
1655 pts/0    00:00:00 sh
1656 pts/0    00:00:00 w3mimgdisplay
1660 pts/0    00:00:00 ps
root@astakhov:~# kill -9 1630
root@astakhov:~# ps
  PID TTY      TIME CMD
 1626 pts/0    00:00:00 bash
 1637 pts/0    00:00:00 w3m
 1645 pts/0    00:00:00 sh
 1646 pts/0    00:00:00 w3mimgdisplay
 1647 pts/0    00:00:00 w3m
 1655 pts/0    00:00:00 sh
 1656 pts/0    00:00:00 w3mimgdisplay
 1662 pts/0    00:00:00 ps
[1] Убито                w3m yandex.ru
```

Рисунок 3 - просмотр процессов

Задание:

6. Вывести список всех процессов всех пользователей
7. Просмотреть список процессов постранично

Практическая часть: Для того чтобы посмотреть список всех процессов постранично введем команду “ps axu | more”

```
root@astakhov:~# ps axu | more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2 36416  9128 ?        Ss   15:58   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    15:58   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   15:58   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   15:58   0:00 [rcu_par_gp]
root         6  0.0  0.0      0     0 ?        I<   15:58   0:00 [kworker/0:0H-k
root         7  0.0  0.0      0     0 ?        I    15:58   0:00 [kworker/u2:0-e
root         8  0.0  0.0      0     0 ?        I<   15:58   0:00 [mm_percpu_wq]
root         9  0.0  0.0      0     0 ?        S    15:58   0:00 [ksoftirqd/0]
root        10  0.0  0.0      0     0 ?        I    15:58   0:00 [rcu_sched]
root        11  0.0  0.0      0     0 ?        T    15:58   0:00 [rcu_bh]
```

Рисунок 4 - постраничный список всех процессов

Задание:

8. Отобрать из вывода команды ps строку, соответствующую процессу «dbus-daemon», определить, где лежит её выполняемый файл и с какими параметрами он запущен.

Практическая часть: для поиска информации о процессе “dbus-daemon” воспользуемся командой “ps axu | grep dbus-daemon”

```
root@astakhov:~# ps axu | grep dbus-daemon
message+  376  0.0  0.0 6872 3624 ?        Ss   15:58   0:00 /usr/bin/dbus-daemon --systemd --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
```

Рисунок 5 - информация о процессе “dbus-daemon”

Задание:

9. Записать в файл с именем, содержащим текущее время, строку «-----»и список процессов.

Практическая часть: Воспользуемся командой “bash -c ‘echo “- - - -” && ps’ > \$(date +%H:%M%S).txt”.

```

root@astakhov:~# bash -c 'echo "-----" && ps' > $(date +%H:%M:%S).txt
root@astakhov:~# ls
17:31:37.txt          images
17:32:52.txt          leafpad-0.8.18.1
2021-09-16-205849_1920x925_screenshot.png leafpad_0.8.18.1.orig.tar.gz
2021-09-17.txt        log.txt
date                  Видео
idesktop              Документы
root@astakhov:~# cat 17*37.txt
-----
  PID TTY          TIME CMD
 1626 pts/0        00:00:00 bash
 1637 pts/0        00:00:00 w3m
 1645 pts/0        00:00:00 sh
 1646 pts/0        00:00:00 w3mimgdisplay
 1647 pts/0        00:00:00 w3m
 1655 pts/0        00:00:00 sh
 1656 pts/0        00:00:00 w3mimgdisplay
 1687 pts/0        00:00:00 ps
root@astakhov:~#

```

Рисунок 6 - запись списка процессов в файл

Задание:

9. Выполнить команду в фоновом режиме с отсрочкой запуска на 1 минуту.

Практическая часть: Воспользуемся командой “(sleep 60 && screenfetch)&”.

```

root@astakhov:~# (sleep 60 && screenfetch)&
[5] 1842
root@astakhov:~#

```

```

_,met$$$$$gg.
,g$$$$$$$$$$$$P.
,g$P""          ""Y$.
,$$P'           $$$
',$$P          ,ggs.  $b:
`d$$'          ,P""   $$$
$$P           d$'     ,$$P
$$:           $$.    - ,d$$'
$$\;          Y$b._   d$P'
Y$$.          "Y$$$$P"'
`$$b          "-._
`Y$$
`Y$$.
`$$b.
`Y$b.
"Y$b.
.....

```

```

OS: Debian 10 buster
Kernel: i686 Linux 4.19.0-14-686-pae
Uptime: 1h 39m
Packages: 868
Shell: bash
Resolution: 1600x900
WM: OpenBox
GTK Theme: Adwaita [GTK3]
Icon Theme: Tango
Disk: 2,4G / 7,8G (32%)
CPU: Intel Core i3-10110U @ 2.592GHz
GPU: InnoTek Systemberatung GmbH Virtu
RAM: 463MiB / 3974MiB

```

Рисунок 7 - запуск команды с отсрочкой в фоновом режиме

Задание:

10. Отобрать из одного из сформированных файлов строки, относящиеся к одному из процессов.

Практическая часть: Для выполнения этой задачи воспользуемся командами “cat” и “grep”.

```

root@astakhov:~# cat 17*37.txt | grep w3m
1637 pts/0      00:00:00 w3m
1646 pts/0      00:00:00 w3mimgdisplay
1647 pts/0      00:00:00 w3m
1656 pts/0      00:00:00 w3mimgdisplay
root@astakhov:~#

```

Рисунок 8 - поиск информации о процессе в файле

Задание:

11. Вывести результаты работы произвольной команды в один файл, а сообщения об ошибках в другой. Продемонстрировать правильность работы.

Практическая часть: Воспользуемся перенаправлением потоков.

```

root@astakhov:~# (echo "task for std streams" && tar -zcvx archive.tar.gz ~/.idesktop/) > log.txt
t 2>err.txt
root@astakhov:~# cat log.txt
task for std streams
root@astakhov:~# cat err.txt
tar: Указать можно только один из параметров «-Acdrtrux», «--delete» или «--test-label»
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
root@astakhov:~#

```

Рисунок 9 - перенаправление потоков

Задание:

12. Выполнить произвольную команду с ограничением использования процессорного времени 300 секунд и выводом результатов и сообщений об ошибках в один файл.

Практическая часть: воспользуемся командой “ulimit” для ограничения ресурсов.

```

root@astakhov:~# bash -c 'ulimit -t 300 && screenfetch' &>log.txt
root@astakhov:~# cat log.txt
      _ ,met$$$$$gg.      root@astakhov
    ,g$$$$$$$$$$$$$$P.    OS: Debian 10 buster
   ,g$$$P""           ""Y$$.". Kernel: i686 Linux 4.19.0-14-686-pae

```

Рисунок 10 - выполнение команды с ограничением ресурсов

Задание:

13. Настроить cron на выполнение команды ежедневно в заданное время. Продемонстрировать правильность работы.

Практическая часть:

```

# m h dom mon dow  command
0 12 * * * screenfetch

root@astakhov:~# history 2
286  crontab -l
287  history 2
root@astakhov:~#

```

Рисунок 11 - просмотр содержимого crontab

Вывод: в ходе данной лабораторной работы были получены базовые навыки работы с процессами и потоками в ОС Debian.