Bash Notes. Linux environment

Antipov Alexander

What is shell. What is Bash

wiki

Use cases

- Process interaction
- Export of metrics
- Basic interface for interaction w/ Linux kernel: configuration, package management

Making printing faster

Key Bindings

Ctrl+A - go to the start of bash text line

Ctrl+E - to the end of line

Ctrl+K - clear line after cursor

Ctrl+U - clear line before cursor

Ctrl+W - delete last word

Ctrl+L - clear screen

Ctrl+R + - recursive search of previous commands (tap **Ctrl+R** for switching results, **ESC** - for exit search)

Alt+F - jump to next word

Alt+B - jump to prev word

copy / past

Ctrl + C/V - windows shell

[Ctrl + Shift + C/V] or [right click / right click] or [mouse wheel click / mouse wheel click] - linux shell

Simple text highlighting / **right click** - supported in MobaXTerm

inline commands & directory navigation

```
# run the last executed command
!!
# run the n-th line in history
!n
# run the last executed command with <keyword>
!<keyword>
# run the last executed command which contains <keyword>
!?<keyword>?

# switch to prev directory
cd -
# switch to home
cd
```

Making search faster

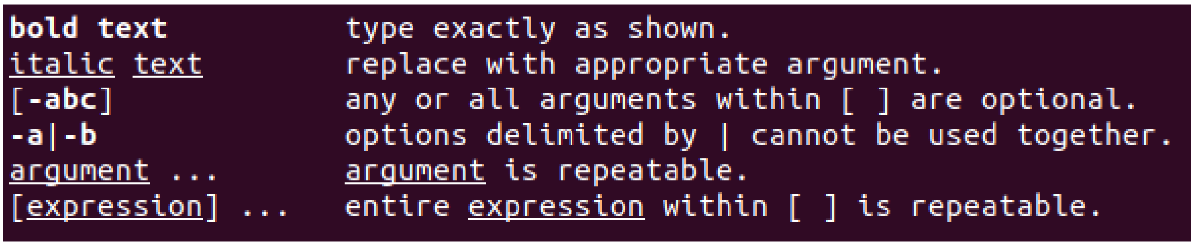
Manual (`man man`)

Manual consists of **9 parts**. It is important to know the description of called command:

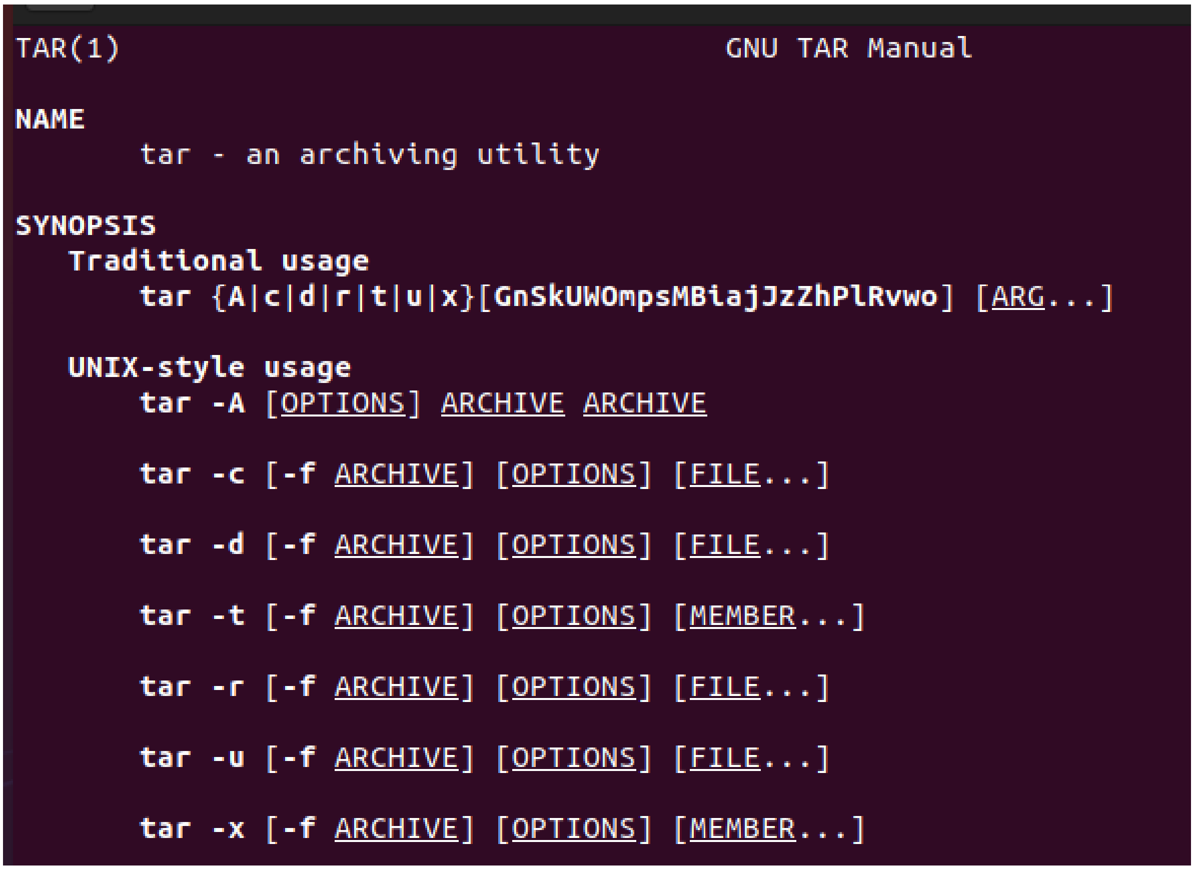
1. -- shell commands (`man 1 <command>`)
2. . -- system calls (`man 2 <command>`)
3. -- libc functions (`man 3 <command>`)

Man pages synopsys

> Upd: Modern alternative of `man` - `info`:
>
> * Supports hyperlinks



```
bold text           type exactly as shown.
italic text         replace with appropriate argument.
[-abc]              any or all arguments within [ ] are optional.
-a|-b               options delimited by | cannot be used together.
argument ...        argument is repeatable.
[expression] ...    entire expression within [ ] is repeatable.
```



```
TAR(1)                          GNU TAR Manual

NAME
       tar - an archiving utility

SYNOPSIS
   Traditional usage
       tar {A|c|d|r|t|u|x}[GnSkUWOmpsMBiajJzZhPlRvwo] [ARG...]

   UNIX-style usage
       tar -A [OPTIONS] ARCHIVE ARCHIVE

       tar -c [-f ARCHIVE] [OPTIONS] [FILE...]

       tar -d [-f ARCHIVE] [OPTIONS] [FILE...]

       tar -t [-f ARCHIVE] [OPTIONS] [MEMBER...]

       tar -r [-f ARCHIVE] [OPTIONS] [FILE...]

       tar -u [-f ARCHIVE] [OPTIONS] [FILE...]

       tar -x [-f ARCHIVE] [OPTIONS] [MEMBER...]
```

- `<elem>` -- elem is mandatory
- `{}` - are used in conjunction with `-a|-b` and not optional

Man navigation

Short command info

```
man -f <command>
whatis <command>
```

Search keyword in a short info

```
man -k <keyword>
```

Arrows, Page Up/Down - page strolling

u / d - half page strolling

`/<key1>|<key2>|<key3>` + **Enter** - regex search of set of keys; **n / N** - switching occurrences (next / previous)

`-N` / `-n` - numeric lines ON / OFF

Simple commands

`which`, `whereis`, `whatis (equvivalent to man -f)`

```
# short info
alexander@alexander-VirtualBox:~/tmp2$ man -f which whereis
which (1)            - locate a command
whereis (1)          - locate the binary, source, and manual page files for a command

# usage
alexander@alexander-VirtualBox:~/tmp2$ which gcc
/usr/bin/gcc
alexander@alexander-VirtualBox:~/tmp2$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/share/gcc /usr/share/man/man1/gcc.1.gz
alexander@alexander-VirtualBox:~/tmp2$ which gucc
alexander@alexander-VirtualBox:~/tmp2$
```

> Instead of using `man -f` type `whatis` command - the same output.

Grep, find

```
find $where_to_find -name $filename_or_dir
grep -r "$match_string" $where_to_find
```

useful find flags

```
-maxdepth
-type
-newer
-name / -wholname
```

useful grep flags

```
rgrep --exclude-dir=[cmake,build] --binary-files=without-match
-E # Extended regex
-e # find pattern
-o # only matching
-n # line number
...
```

About logging

History (`man history`)

Bash command history. Contains last `HISTSIZE` commands (see `~/.bashrc` file) of terminal session/

Append <= `HISTSIZE` lines to the file `~/.bash_history`:

```
history -a
```

Clear local history:

```
history -c
```

Ignore duplicates:

```
# in ~/.bashrc:
HISTCONTROL=ignoreboth:erasedups
```

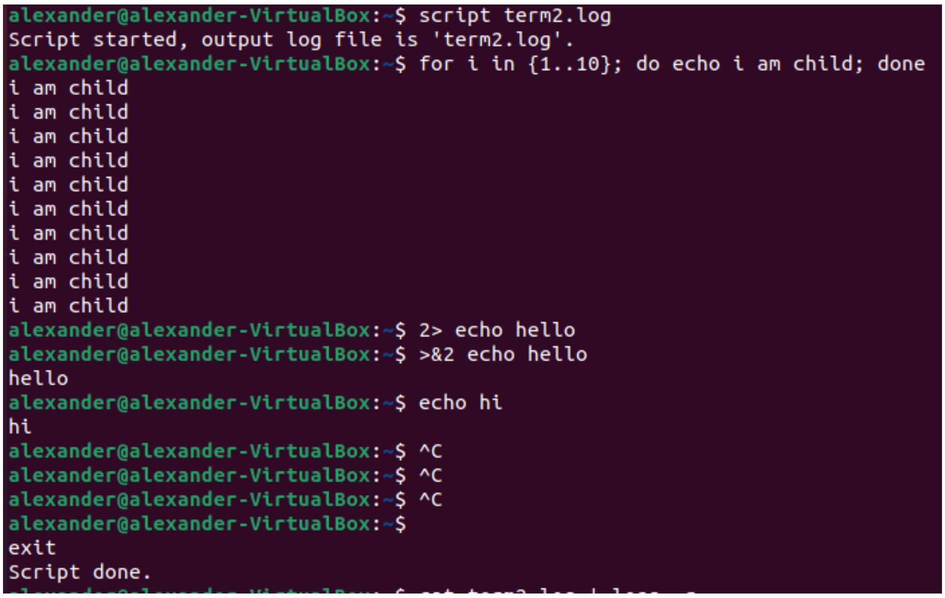
Ctrl+R + - recursive search of previous commands (tap **Cntrl+R** for switching results, **ESC** - for exit search)

Script (`man script`)

Creates **child session** and store **all commands and output of terminal session** into file:

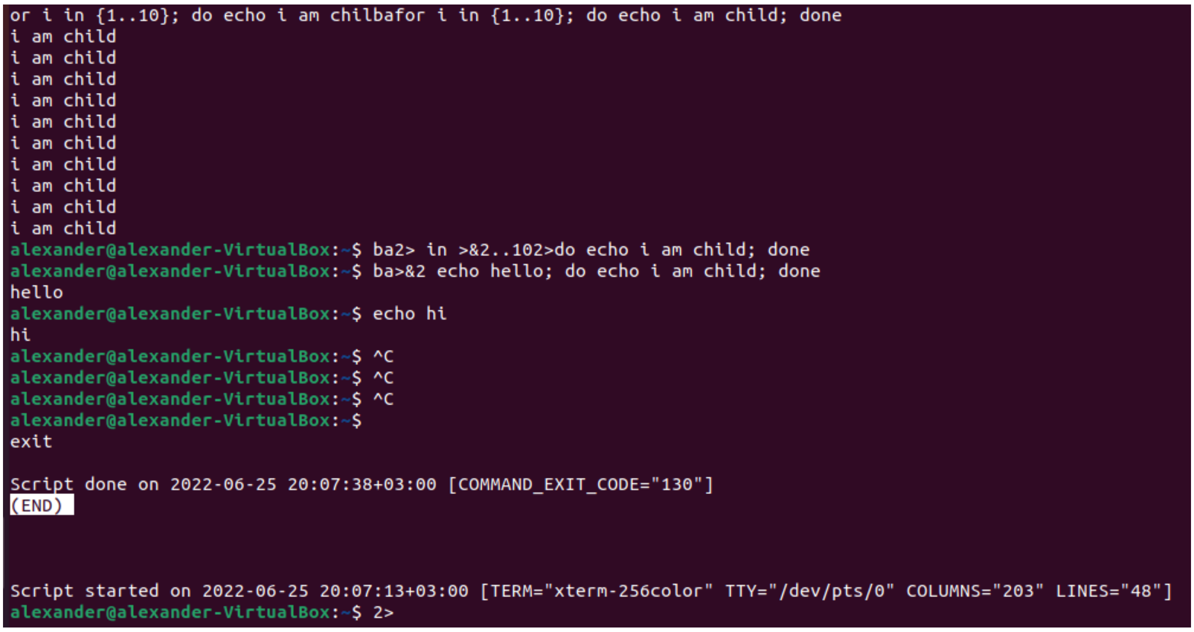
```
script terminal.log
```

Use `less -r` to see it. But there can be some conflicts with **control characters**. That's why you can use vim too.



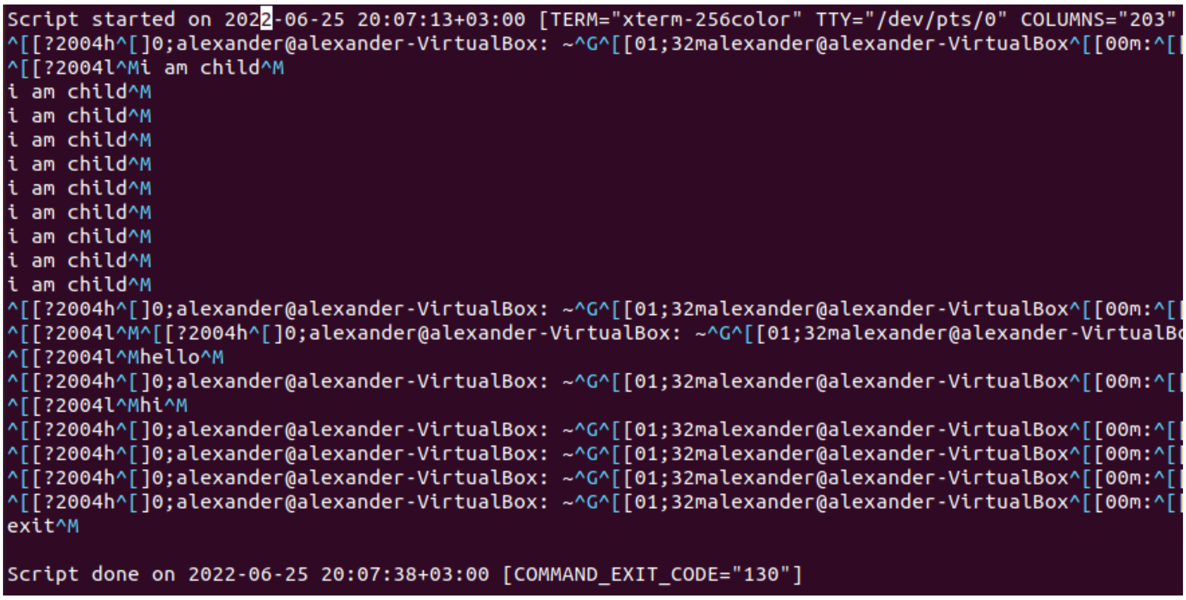
```
alexander@alexander-VirtualBox:~$ script term2.log
Script started, output log file is 'term2.log'.
alexander@alexander-VirtualBox:~$ for i in {1..10}; do echo i am child; done
i am child
i am child
i am child
i am child
i am child
i am child
i am child
i am child
i am child
i am child
alexander@alexander-VirtualBox:~$ 2> echo hello
alexander@alexander-VirtualBox:~$ >&2 echo hello
hello
alexander@alexander-VirtualBox:~$ echo hi
hi
alexander@alexander-VirtualBox:~$ ^C
alexander@alexander-VirtualBox:~$ ^C
alexander@alexander-VirtualBox:~$ ^C
alexander@alexander-VirtualBox:~$
exit
Script done.
```

```
cat term2.log | less -r
```



```
or i in {1..10}; do echo i am chilbafor i in {1..10}; do echo i am child; done
i am child
i am child
i am child
i am child
i am child
i am child
i am child
i am child
i am child
i am child
alexander@alexander-VirtualBox:~$ ba2> in >&2..102>do echo i am child; done
alexander@alexander-VirtualBox:~$ ba>&2 echo hello; do echo i am child; done
hello
alexander@alexander-VirtualBox:~$ echo hi
hi
alexander@alexander-VirtualBox:~$ ^C
alexander@alexander-VirtualBox:~$ ^C
alexander@alexander-VirtualBox:~$ ^C
alexander@alexander-VirtualBox:~$
exit

Script done on 2022-06-25 20:07:38+03:00 [COMMAND_EXIT_CODE="130"]
(END)



Script started on 2022-06-25 20:07:13+03:00 [TERM="xterm-256color" TTY="/dev/pts/0" COLUMNS="203" LINES="48"]
alexander@alexander-VirtualBox:~$ 2>
```

```
vim term2.log
```



```
Script started on 2022-06-25 20:07:13+03:00 [TERM="xterm-256color" TTY="/dev/pts/0" COLUMNS="203"
^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBox^[[00m:^[
^[[?2004l^Mi am child^M
i am child^M
i am child^M
i am child^M
i am child^M
i am child^M
i am child^M
i am child^M
i am child^M
i am child^M
^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBox^[[00m:^[
^[[?2004l^M^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBo
^[[?2004l^Mhello^M
^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBox^[[00m:^[
^[[?2004l^Mhi^M
^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBox^[[00m:^[
^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBox^[[00m:^[
^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBox^[[00m:^[
^[[?2004h^[]0;alexander@alexander-VirtualBox: ~^G^[[01;32malexander@alexander-VirtualBox^[[00m:^[
exit^M

Script done on 2022-06-25 20:07:38+03:00 [COMMAND_EXIT_CODE="130"]
```

Bash Environment (`man bash`)

Local variables

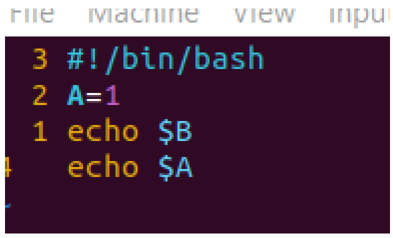
Local variables -- variables that are present within the current instance of the shell (or function). Not available for programs started in a child shell session.

```
# print all local variables
set | less
# set new one in a shell or shell script
SOMEVAR="some value string"
# print it
echo $SOMEVAR
# delete declarated var
unset SOMEVAR
```

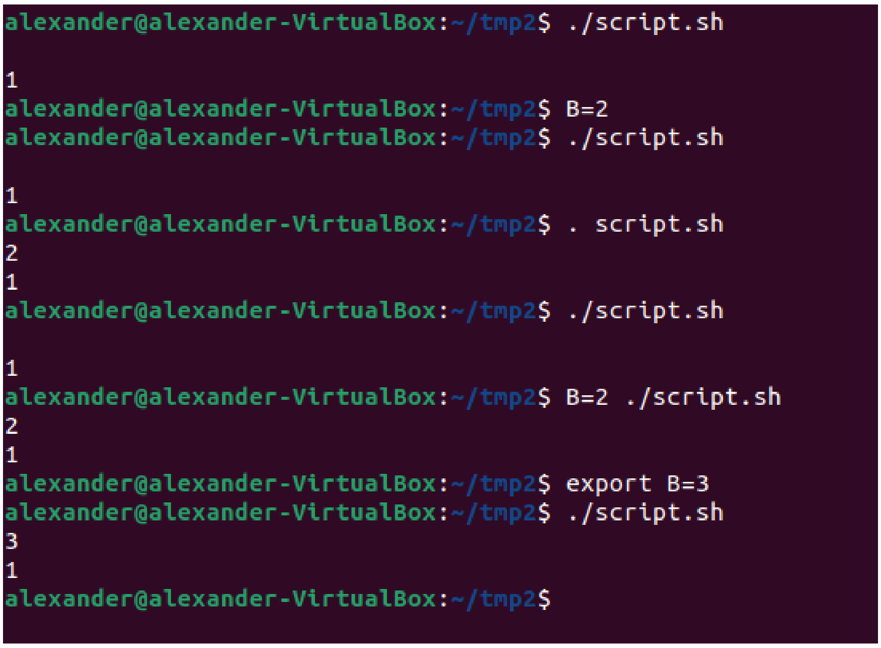
Local variables are used for transfering variables to the process only, for example:

```
CC=/opt/riscv32/bin/riscv32-unknown-elf-gcc ../configure --host=/opt/riscv32
```

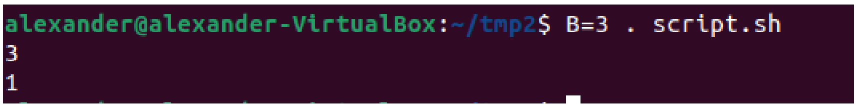
Environment data in C sources can be accessed via `envp` (`argc`, `argv`, `evnp` - 3 arguments of `main` function)



```
#!/bin/bash
A=1
echo $B
echo $A
```



```
alexander@alexander-VirtualBox:~/tmp2$ ./script.sh

1
alexander@alexander-VirtualBox:~/tmp2$ B=2
alexander@alexander-VirtualBox:~/tmp2$ ./script.sh

1
alexander@alexander-VirtualBox:~/tmp2$ . script.sh
2
1
alexander@alexander-VirtualBox:~/tmp2$ ./script.sh

1
alexander@alexander-VirtualBox:~/tmp2$ B=2 ./script.sh
2
1
alexander@alexander-VirtualBox:~/tmp2$ export B=3
alexander@alexander-VirtualBox:~/tmp2$ ./script.sh
3
1
alexander@alexander-VirtualBox:~/tmp2$
```



```
alexander@alexander-VirtualBox:~/tmp2$ B=3 . script.sh
3
1
```

Environment variables

Unlike local variables these variables can be accessed via child shell sessions.

```
# print available environment
env | less
# declaration of new envrionment var
export ENVAR="3"
# delete it
export -n ENVAR
```

> In bash manual there is a definition of shell variables - the 3rd group of variables used for shell runtime. They can be either local or global.

```
# export PATH=$PATH:<dir>
$PATH
$CMD
$PWD


```

Making your own bash environment

It is better to use private shell script (`my_env.sh` for instance) for setting up environment on the start of the shell. For this purpose make `chmod +x my_env.sh`, add its path to `~/.bashrc`.

> `~/.bashrc` executes on the start of running new local shell session
>
> `~/.profile`, `~/.bash_profile` execute on the start of SSH session and include `~/.bashrc`.

Special variables

Many of them are used in shell scripts, but ones can be used in shell command prompt.

```
# Exit status or status of last executed command (most popular)
echo $?
```

```
# Filename of current shell script
echo $0
# n = 1, 2, ... - positional argumnts of shell script
echo ${1} ${2}
# The number of arguments supplied to a script
echo $#
# PID of current shell
echo $$
# PID of last background command
echo $!
# string of all arguments: "arg1 arg2 arg3"
echo $*
# list of all arguments: "arg1" "arg2" "arg3"
echo $@
```

Aliases

Allows to rename frequently used bash commands to short prints. Default aliases are located in `~/.bashrc`. To add new one it is recommended way to store it into `~/.bash_aliases`.

Print all available aliases:

```
alias
```

Creating & deleting alias:

```
# create alias named word for <command>
alias word=`<command>`
# delete word alias
unalias word
```

Example:

```
# filtered text search
alias grep="grep -r --exclude-dir=[cmake,build] --binary-files=without-match"
```

> `alias rm="rm -i"` will work incorrectly w/ `rm -rf <filedir>`. Why?

Command pipeline

Redirect stdout of the 1st command to the stdin of the next command:

```
cat trace_log.txd | grep "addi" | less -S
```

Pipelined execution:

```
# <cmd2> will run only if <cmd1> has executed correctly
<cmd1> && <cmd2>
# <cmd2> will run after <cmd1>, exit status of <cmd1> is ignored
<cmd1> ; <cmd2>

# what does it mean?
<cmd1> || <cmd2>
```

Example:

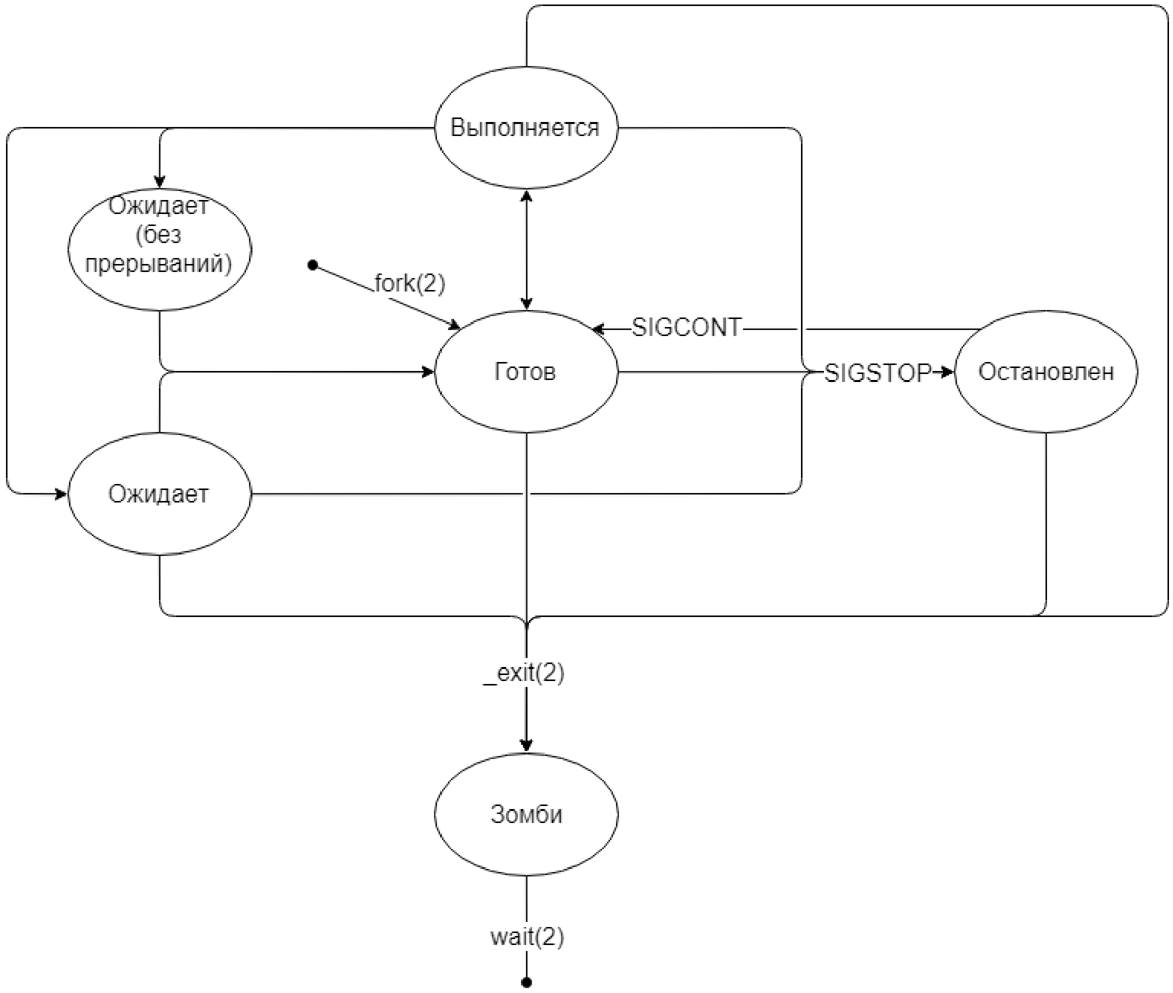
```
traceGen.py > log.txt && process.py log.txt > result.csv && echo $?
```

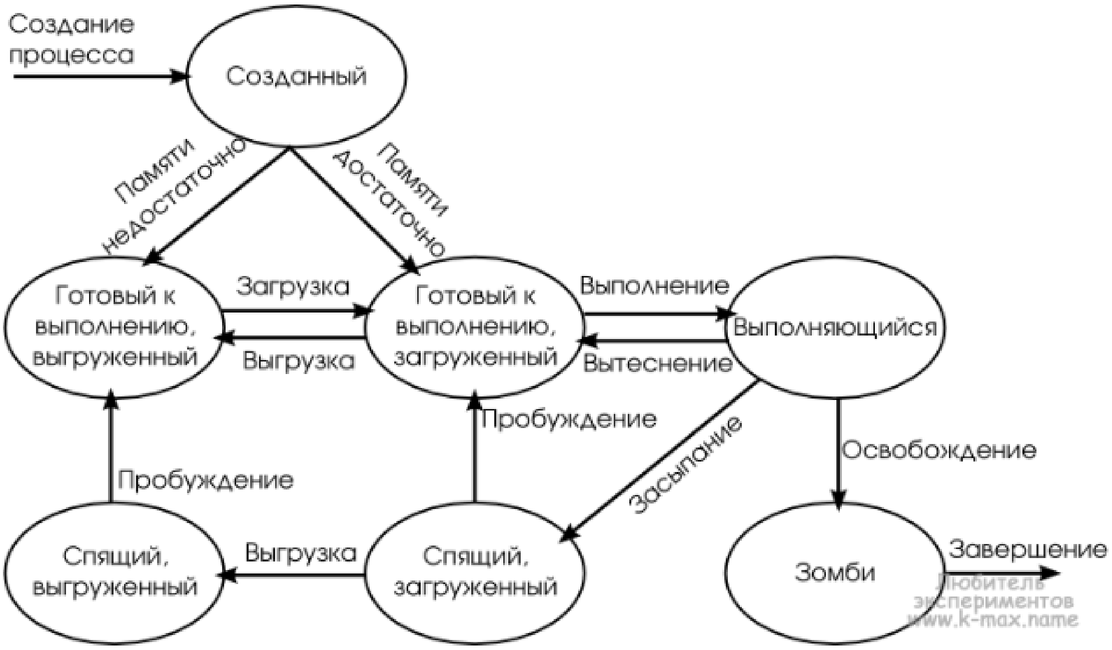
File stream redirections

```
# redirect stdout of <cmd> to the <file>
<cmd> > <file>
# redirect stdout of <cmd> to the file and APPEND data
<cmd> >> <file>
# redirect stderr
<cmd> 2> <file>
# redirect both stdout, stderr to the <file>
<cmd> &> <file>

# redirect both input and output from/to file
program.out < input.log > output.log
```

Process interaction

> Upd: "Роберт Лав. Ядро Линукс. Описание процесса разработки"





process states and livecycle

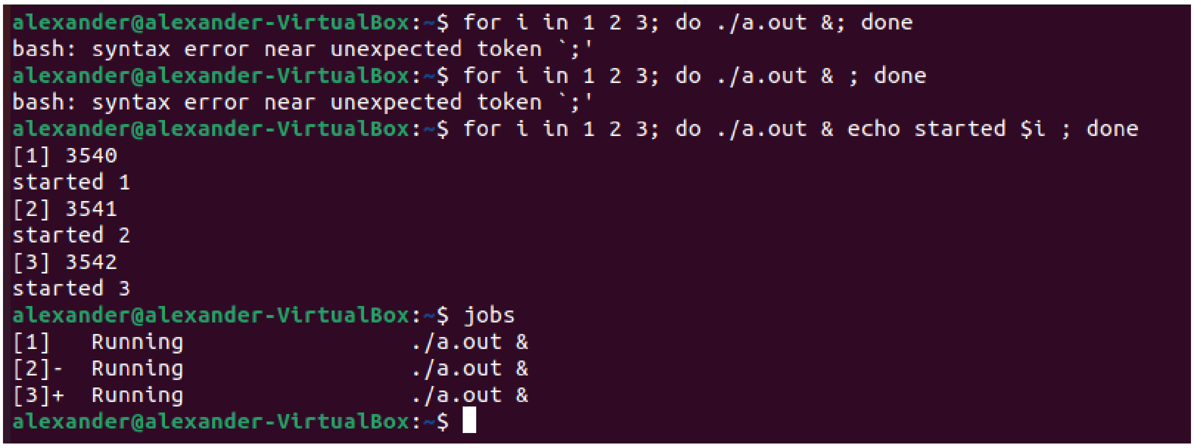
```
# process states
R  Running
S  Sleeping in an interruptible wait
D  Waiting in uninterruptible disk sleep
Z  Zombie
T  Stopped (on a signal) or (before Linux 2.6.33) trace stopped
t  Tracing stop (Linux 2.6.33 onward)
W  Paging (only before Linux 2.6.0)
X  Dead (from Linux 2.6.0 onward)
x  Dead (Linux 2.6.33 to 3.13 only)
K  Wakekill (Linux 2.6.33 to 3.13 only)
W  Waking (Linux 2.6.33 to 3.13 only)
P  Parked (Linux 3.9 to 3.13 only)
```

Foreground vs background processes

`sudo apt install htop` for installing simple and pretty inline task manager where all commands and keys bindings are subscribed.

But in many cases the number of background processes run by your own do not exceed 10 and can be simply managed as jobs:

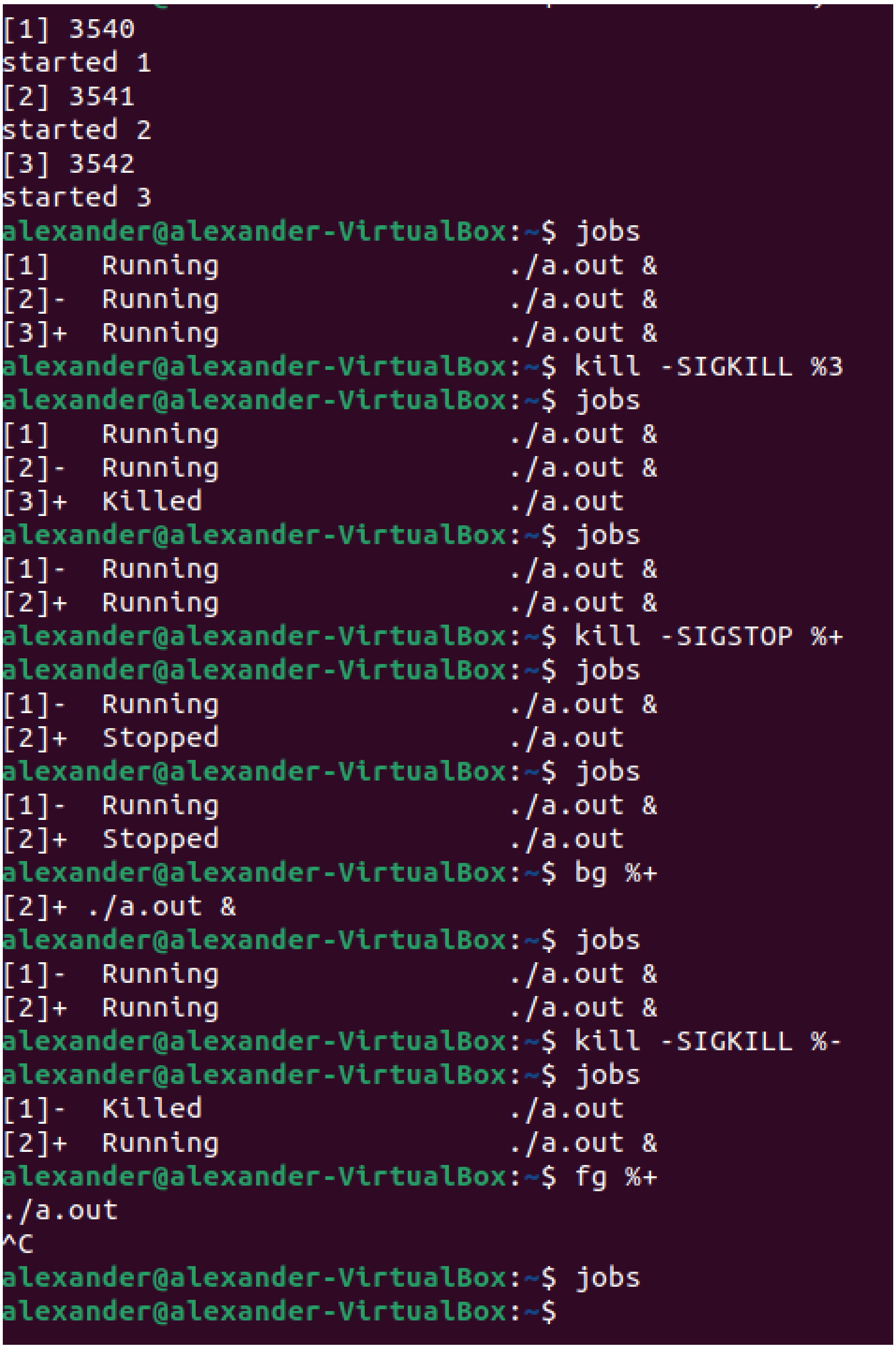
```
jobs
```



```
alexander@alexander-VirtualBox:~$ for i in 1 2 3; do ./a.out &; done
bash: syntax error near unexpected token `;'
alexander@alexander-VirtualBox:~$ for i in 1 2 3; do ./a.out & ; done
bash: syntax error near unexpected token `;'
alexander@alexander-VirtualBox:~$ for i in 1 2 3; do ./a.out & echo started $i ; done
[1] 3540
started 1
[2] 3541
started 2
[3] 3542
started 3
alexander@alexander-VirtualBox:~$ jobs
[1]   Running                 ./a.out &
[2]-  Running                 ./a.out &
[3]+  Running                 ./a.out &
alexander@alexander-VirtualBox:~$ 
```

Return background process to the foreground:

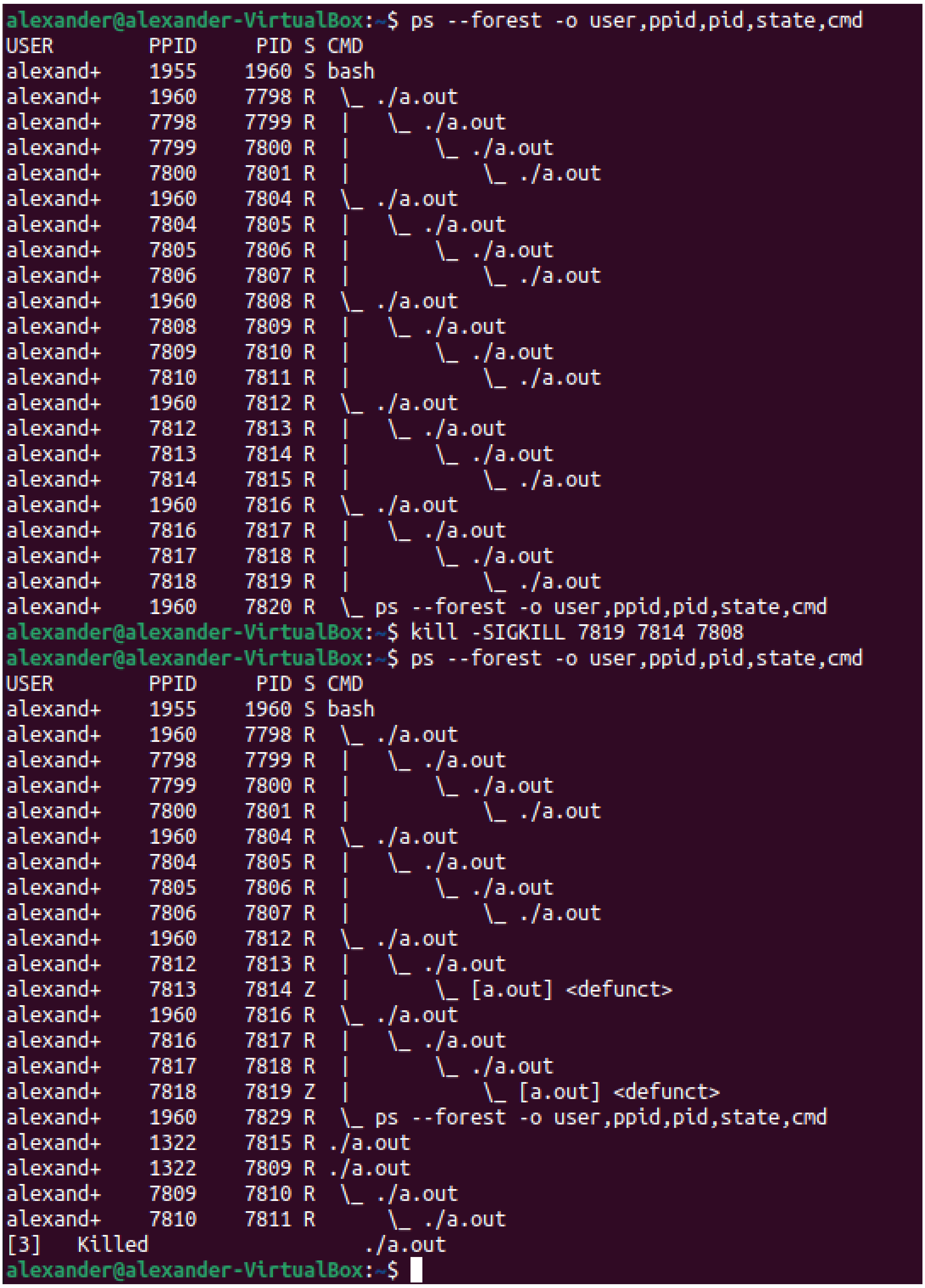
```
# return process [n] to the foreground
fg %n
# %+ / %- are last / previous jobs
# return process [n] to the background
bg %n
```



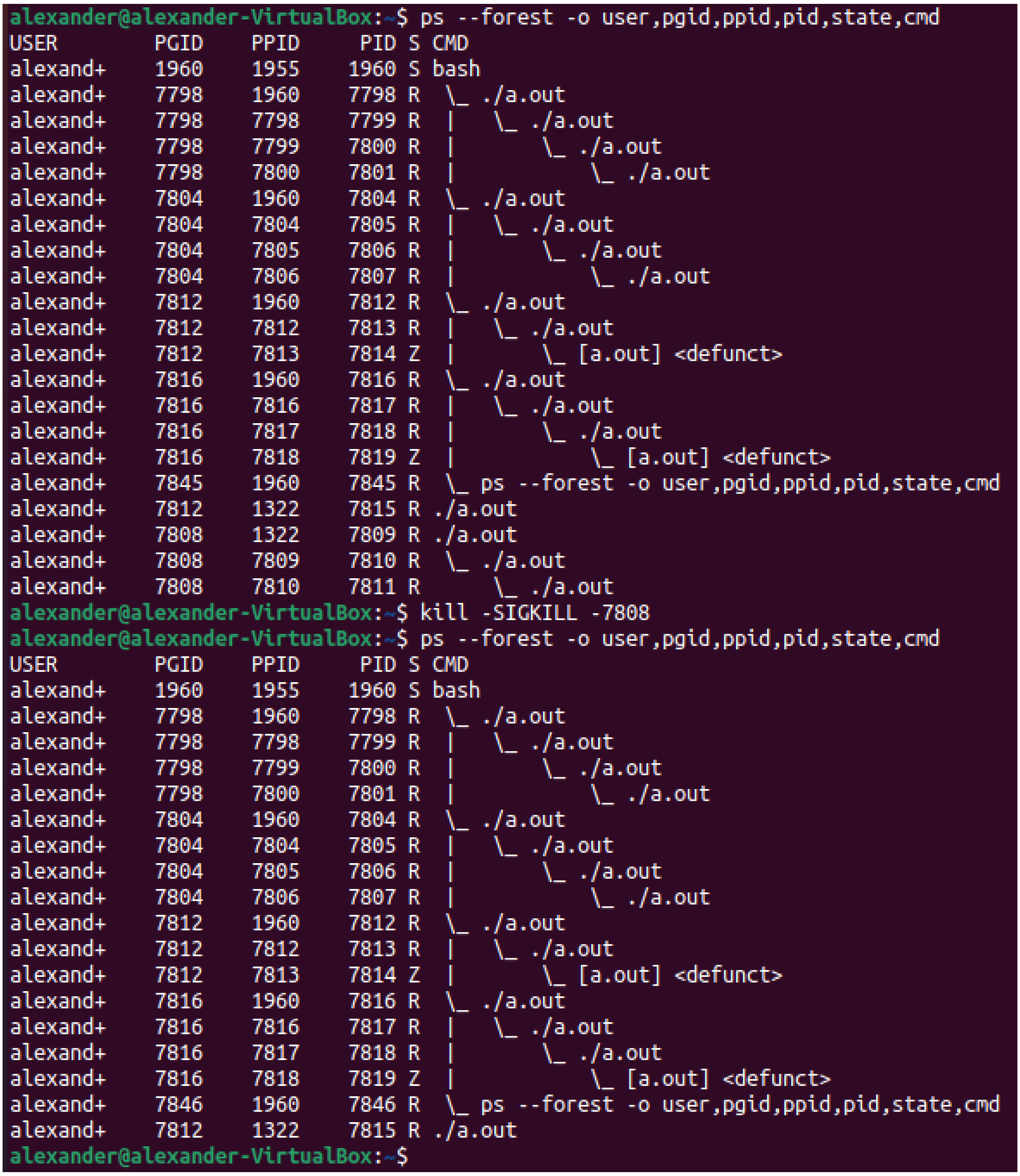
```
[1] 3540
started 1
[2] 3541
started 2
[3] 3542
started 3
alexander@alexander-VirtualBox:~$ jobs
[1]   Running                 ./a.out &
[2]-  Running                 ./a.out &
[3]+  Running                 ./a.out &
alexander@alexander-VirtualBox:~$ kill -SIGKILL %3
alexander@alexander-VirtualBox:~$ jobs
[1]   Running                 ./a.out &
[2]-  Running                 ./a.out &
[3]+  Killed                  ./a.out
alexander@alexander-VirtualBox:~$ jobs
[1]-  Running                 ./a.out &
[2]+  Running                 ./a.out &
alexander@alexander-VirtualBox:~$ kill -SIGSTOP %+
alexander@alexander-VirtualBox:~$ jobs
[1]-  Running                 ./a.out &
[2]+  Stopped                 ./a.out
alexander@alexander-VirtualBox:~$ jobs
[1]-  Running                 ./a.out &
[2]+  Stopped                 ./a.out
alexander@alexander-VirtualBox:~$ bg %+
[2]+ ./a.out &
alexander@alexander-VirtualBox:~$ jobs
[1]-  Running                 ./a.out &
[2]+  Running                 ./a.out &
alexander@alexander-VirtualBox:~$ kill -SIGKILL %-
alexander@alexander-VirtualBox:~$ jobs
[1]-  Killed                  ./a.out
[2]+  Running                 ./a.out &
alexander@alexander-VirtualBox:~$ fg %+
./a.out
^C
alexander@alexander-VirtualBox:~$ jobs
alexander@alexander-VirtualBox:~$
```

> Describe the log.

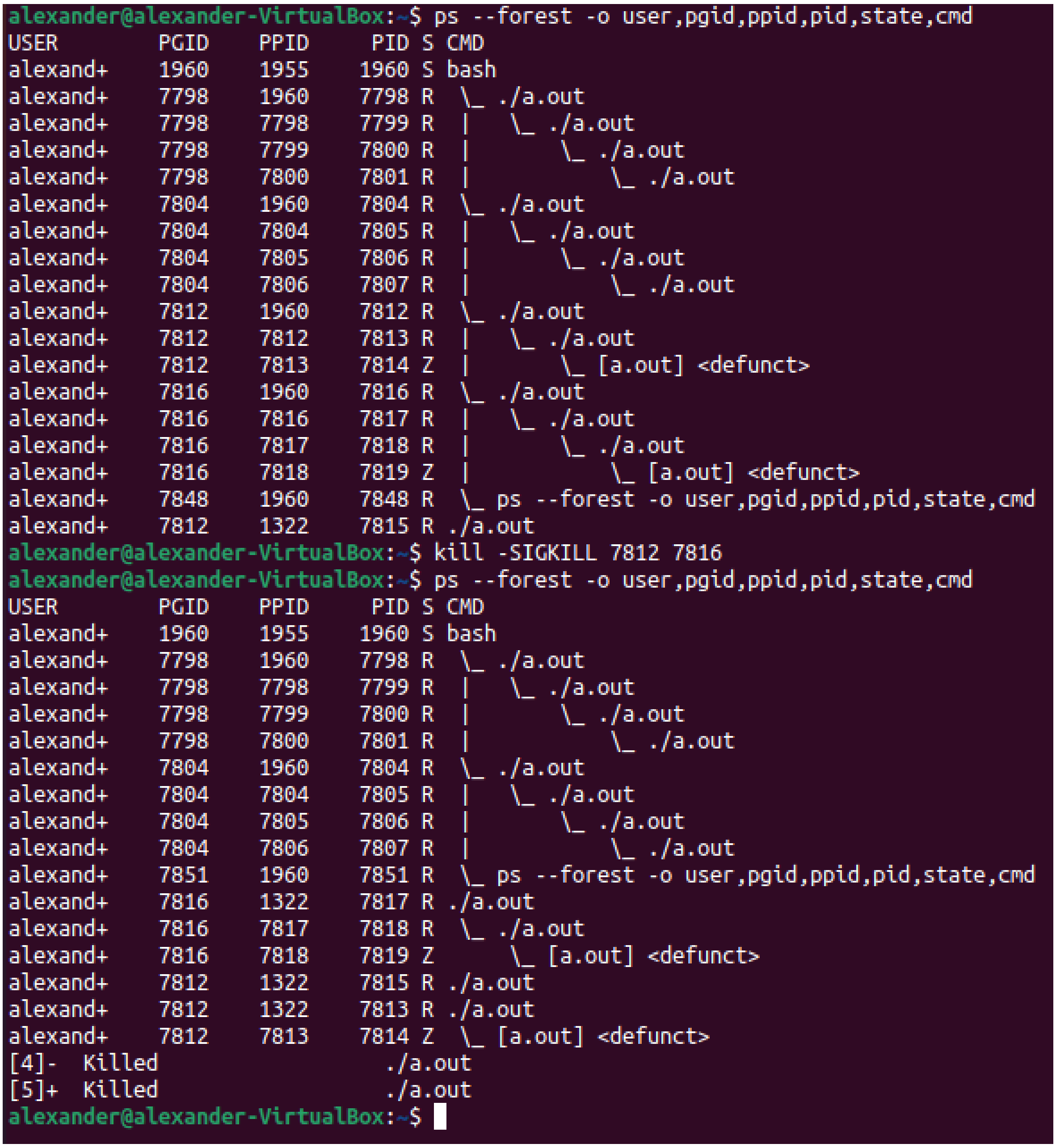
Professional killer



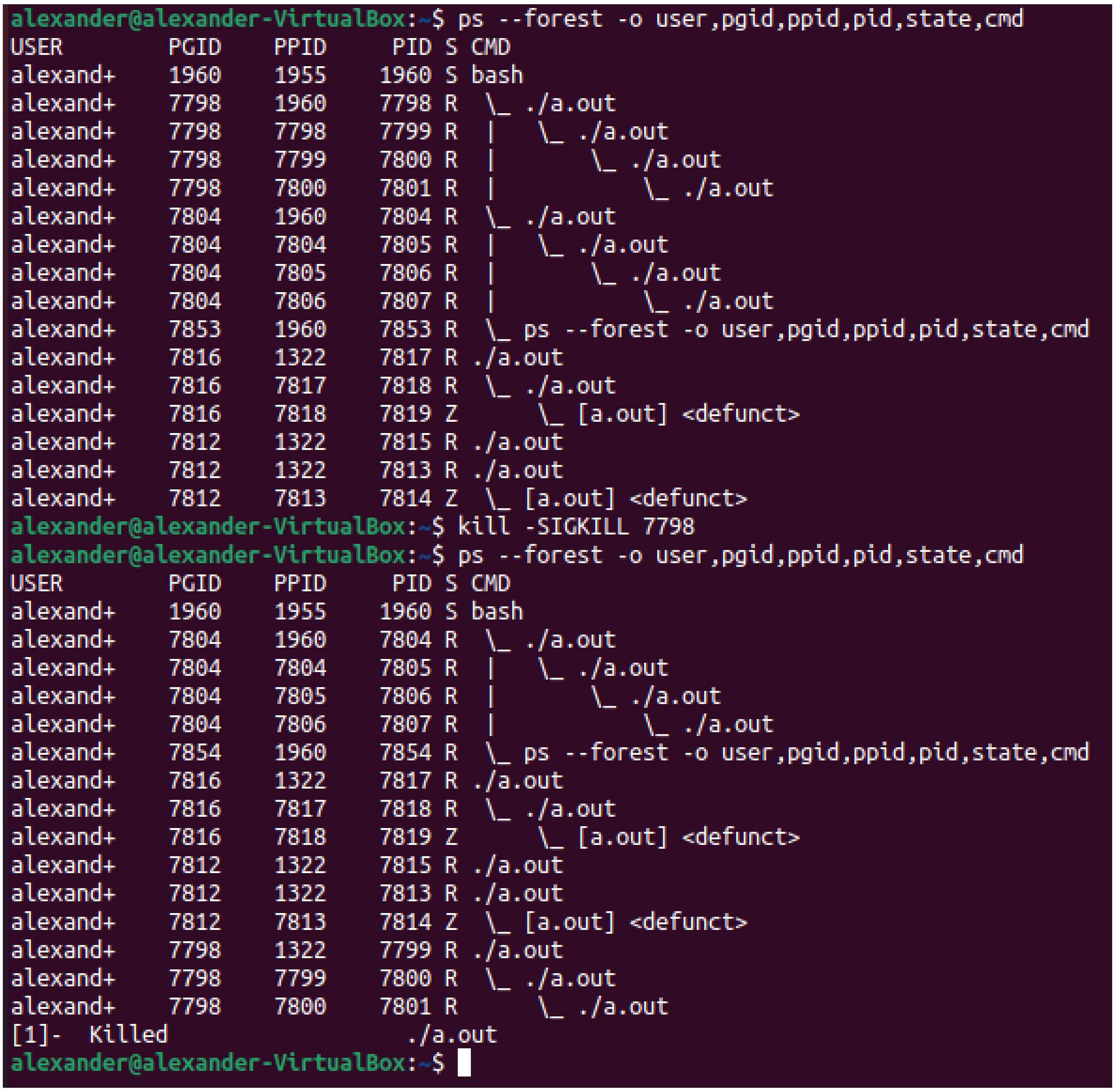
```
alexander@alexander-VirtualBox:~$ ps --forest -o user,ppid,pid,state,cmd
USER        PPID     PID S CMD
alexand+    1955    1960 S bash
alexand+    1960    7798 R  \_ ./a.out
alexand+    7798    7799 R  |   \_ ./a.out
alexand+    7799    7800 R  |       \_ ./a.out
alexand+    7800    7801 R  |           \_ ./a.out
alexand+    1960    7804 R  \_ ./a.out
alexand+    7804    7805 R  |   \_ ./a.out
alexand+    7805    7806 R  |       \_ ./a.out
alexand+    7806    7807 R  |           \_ ./a.out
alexand+    1960    7808 R  \_ ./a.out
alexand+    7808    7809 R  |   \_ ./a.out
alexand+    7809    7810 R  |       \_ ./a.out
alexand+    7810    7811 R  |           \_ ./a.out
alexand+    1960    7812 R  \_ ./a.out
alexand+    7812    7813 R  |   \_ ./a.out
alexand+    7813    7814 R  |       \_ ./a.out
alexand+    7814    7815 R  |           \_ ./a.out
alexand+    1960    7816 R  \_ ./a.out
alexand+    7816    7817 R  |   \_ ./a.out
alexand+    7817    7818 R  |       \_ ./a.out
alexand+    7818    7819 R  |           \_ ./a.out
alexand+    1960    7820 R  \_ ps --forest -o user,ppid,pid,state,cmd
alexander@alexander-VirtualBox:~$ kill -SIGKILL 7819 7814 7808
alexander@alexander-VirtualBox:~$ ps --forest -o user,ppid,pid,state,cmd
USER        PPID     PID S CMD
alexand+    1955    1960 S bash
alexand+    1960    7798 R  \_ ./a.out
alexand+    7798    7799 R  |   \_ ./a.out
alexand+    7799    7800 R  |       \_ ./a.out
alexand+    7800    7801 R  |           \_ ./a.out
alexand+    1960    7804 R  \_ ./a.out
alexand+    7804    7805 R  |   \_ ./a.out
alexand+    7805    7806 R  |       \_ ./a.out
alexand+    7806    7807 R  |           \_ ./a.out
alexand+    1960    7812 R  \_ ./a.out
alexand+    7812    7813 R  |   \_ ./a.out
alexand+    7813    7814 Z  |       \_ [a.out] <defunct>
alexand+    1960    7816 R  \_ ./a.out
alexand+    7816    7817 R  |   \_ ./a.out
alexand+    7817    7818 R  |       \_ ./a.out
alexand+    7818    7819 Z  |           \_ [a.out] <defunct>
alexand+    1960    7829 R  \_ ps --forest -o user,ppid,pid,state,cmd
alexand+    1322    7815 R ./a.out
alexand+    1322    7809 R ./a.out
alexand+    7809    7810 R  \_ ./a.out
alexand+    7810    7811 R      \_ ./a.out
[3]   Killed                  ./a.out
alexander@alexander-VirtualBox:~$ 
```



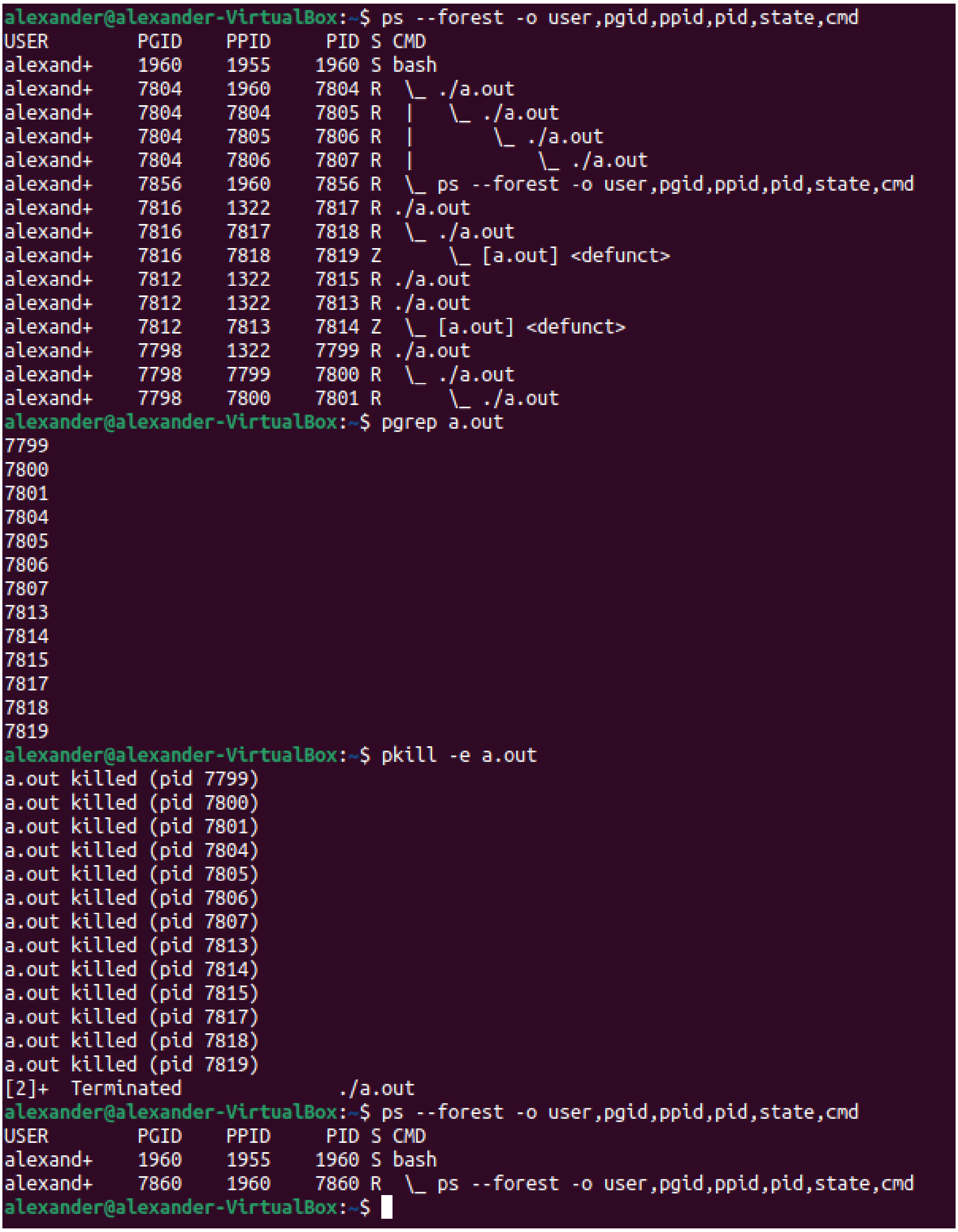
```
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7798    1960    7798 R  \_ ./a.out
alexand+    7798    7798    7799 R  |   \_ ./a.out
alexand+    7798    7799    7800 R  |       \_ ./a.out
alexand+    7798    7800    7801 R  |           \_ ./a.out
alexand+    7804    1960    7804 R  \_ ./a.out
alexand+    7804    7804    7805 R  |   \_ ./a.out
alexand+    7804    7805    7806 R  |       \_ ./a.out
alexand+    7804    7806    7807 R  |           \_ ./a.out
alexand+    7812    1960    7812 R  \_ ./a.out
alexand+    7812    7812    7813 R  |   \_ ./a.out
alexand+    7812    7813    7814 Z  |       \_ [a.out] <defunct>
alexand+    7816    1960    7816 R  \_ ./a.out
alexand+    7816    7816    7817 R  |   \_ ./a.out
alexand+    7816    7817    7818 R  |       \_ ./a.out
alexand+    7816    7818    7819 Z  |           \_ [a.out] <defunct>
alexand+    7845    1960    7845 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexand+    7812    1322    7815 R ./a.out
alexand+    7808    1322    7809 R ./a.out
alexand+    7808    7809    7810 R  \_ ./a.out
alexand+    7808    7810    7811 R      \_ ./a.out
alexander@alexander-VirtualBox:~$ kill -SIGKILL -7808
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7798    1960    7798 R  \_ ./a.out
alexand+    7798    7798    7799 R  |   \_ ./a.out
alexand+    7798    7799    7800 R  |       \_ ./a.out
alexand+    7798    7800    7801 R  |           \_ ./a.out
alexand+    7804    1960    7804 R  \_ ./a.out
alexand+    7804    7804    7805 R  |   \_ ./a.out
alexand+    7804    7805    7806 R  |       \_ ./a.out
alexand+    7804    7806    7807 R  |           \_ ./a.out
alexand+    7812    1960    7812 R  \_ ./a.out
alexand+    7812    7812    7813 R  |   \_ ./a.out
alexand+    7812    7813    7814 Z  |       \_ [a.out] <defunct>
alexand+    7816    1960    7816 R  \_ ./a.out
alexand+    7816    7816    7817 R  |   \_ ./a.out
alexand+    7816    7817    7818 R  |       \_ ./a.out
alexand+    7816    7818    7819 Z  |           \_ [a.out] <defunct>
alexand+    7846    1960    7846 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexand+    7812    1322    7815 R ./a.out
alexander@alexander-VirtualBox:~$
```



```
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7798    1960    7798 R  \_ ./a.out
alexand+    7798    7798    7799 R  |   \_ ./a.out
alexand+    7798    7799    7800 R  |       \_ ./a.out
alexand+    7798    7800    7801 R  |           \_ ./a.out
alexand+    7804    1960    7804 R  \_ ./a.out
alexand+    7804    7804    7805 R  |   \_ ./a.out
alexand+    7804    7805    7806 R  |       \_ ./a.out
alexand+    7804    7806    7807 R  |           \_ ./a.out
alexand+    7812    1960    7812 R  \_ ./a.out
alexand+    7812    7812    7813 R  |   \_ ./a.out
alexand+    7812    7813    7814 Z  |       \_ [a.out] <defunct>
alexand+    7816    1960    7816 R  \_ ./a.out
alexand+    7816    7816    7817 R  |   \_ ./a.out
alexand+    7816    7817    7818 R  |       \_ ./a.out
alexand+    7816    7818    7819 Z  |           \_ [a.out] <defunct>
alexand+    7848    1960    7848 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexand+    7812    1322    7815 R ./a.out
alexander@alexander-VirtualBox:~$ kill -SIGKILL 7812 7816
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7798    1960    7798 R  \_ ./a.out
alexand+    7798    7798    7799 R  |   \_ ./a.out
alexand+    7798    7799    7800 R  |       \_ ./a.out
alexand+    7798    7800    7801 R  |           \_ ./a.out
alexand+    7804    1960    7804 R  \_ ./a.out
alexand+    7804    7804    7805 R  |   \_ ./a.out
alexand+    7804    7805    7806 R  |       \_ ./a.out
alexand+    7804    7806    7807 R  |           \_ ./a.out
alexand+    7851    1960    7851 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexand+    7816    1322    7817 R ./a.out
alexand+    7816    7817    7818 R  \_ ./a.out
alexand+    7816    7818    7819 Z      \_ [a.out] <defunct>
alexand+    7812    1322    7815 R ./a.out
alexand+    7812    1322    7813 R ./a.out
alexand+    7812    7813    7814 Z  \_ [a.out] <defunct>
[4]-  Killed                  ./a.out
[5]+  Killed                  ./a.out
alexander@alexander-VirtualBox:~$ 
```

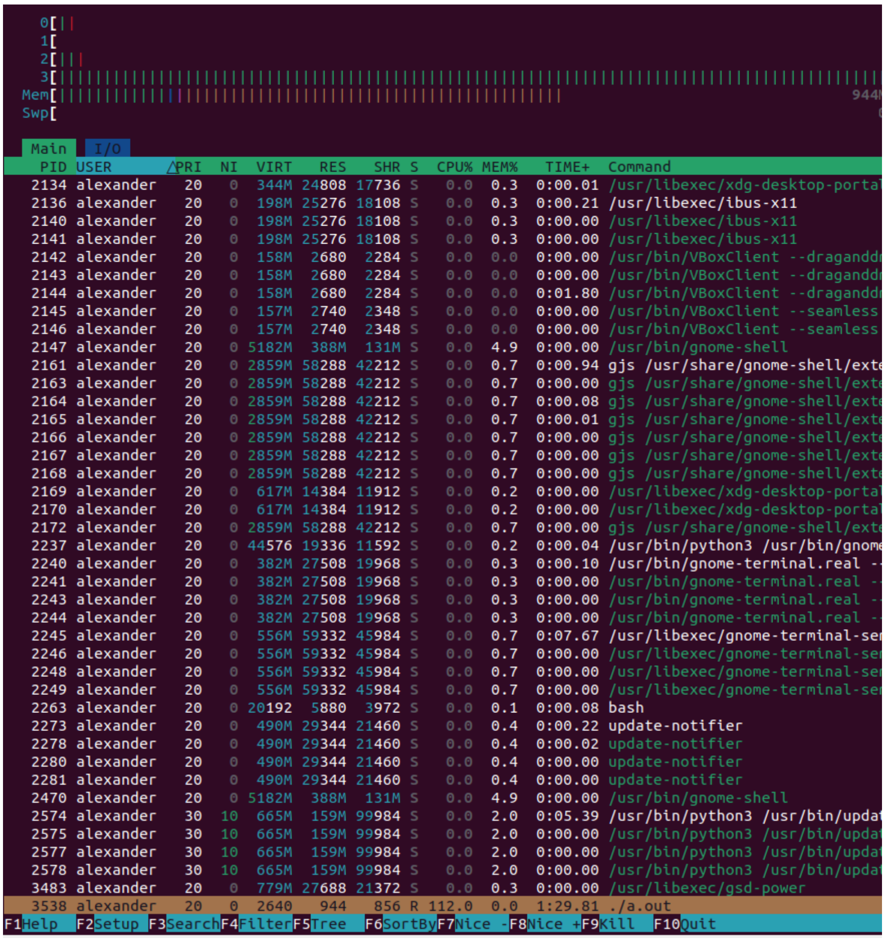


```
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7798    1960    7798 R  \_ ./a.out
alexand+    7798    7798    7799 R  |   \_ ./a.out
alexand+    7798    7799    7800 R  |       \_ ./a.out
alexand+    7798    7800    7801 R  |           \_ ./a.out
alexand+    7804    1960    7804 R  \_ ./a.out
alexand+    7804    7804    7805 R  |   \_ ./a.out
alexand+    7804    7805    7806 R  |       \_ ./a.out
alexand+    7804    7806    7807 R  |           \_ ./a.out
alexand+    7853    1960    7853 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexand+    7816    1322    7817 R ./a.out
alexand+    7816    7817    7818 R  \_ ./a.out
alexand+    7816    7818    7819 Z      \_ [a.out] <defunct>
alexand+    7812    1322    7815 R ./a.out
alexand+    7812    1322    7813 R ./a.out
alexand+    7812    7813    7814 Z  \_ [a.out] <defunct>
alexander@alexander-VirtualBox:~$ kill -SIGKILL 7798
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7804    1960    7804 R  \_ ./a.out
alexand+    7804    7804    7805 R  |   \_ ./a.out
alexand+    7804    7805    7806 R  |       \_ ./a.out
alexand+    7804    7806    7807 R  |           \_ ./a.out
alexand+    7854    1960    7854 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexand+    7816    1322    7817 R ./a.out
alexand+    7816    7817    7818 R  \_ ./a.out
alexand+    7816    7818    7819 Z      \_ [a.out] <defunct>
alexand+    7812    1322    7815 R ./a.out
alexand+    7812    1322    7813 R ./a.out
alexand+    7812    7813    7814 Z  \_ [a.out] <defunct>
alexand+    7798    1322    7799 R ./a.out
alexand+    7798    7799    7800 R  \_ ./a.out
alexand+    7798    7800    7801 R      \_ ./a.out
[1]-  Killed                  ./a.out
alexander@alexander-VirtualBox:~$ 
```



```
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7804    1960    7804 R  \_ ./a.out
alexand+    7804    7804    7805 R  |   \_ ./a.out
alexand+    7804    7805    7806 R  |       \_ ./a.out
alexand+    7804    7806    7807 R  |           \_ ./a.out
alexand+    7856    1960    7856 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexand+    7816    1322    7817 R ./a.out
alexand+    7816    7817    7818 R  \_ ./a.out
alexand+    7816    7818    7819 Z      \_ [a.out] <defunct>
alexand+    7812    1322    7815 R ./a.out
alexand+    7812    1322    7813 R ./a.out
alexand+    7812    7813    7814 Z  \_ [a.out] <defunct>
alexand+    7798    1322    7799 R ./a.out
alexand+    7798    7799    7800 R  \_ ./a.out
alexand+    7798    7800    7801 R      \_ ./a.out
alexander@alexander-VirtualBox:~$ pgrep a.out
7799
7800
7801
7804
7805
7806
7807
7813
7814
7815
7817
7818
7819
alexander@alexander-VirtualBox:~$ pkill -e a.out
a.out killed (pid 7799)
a.out killed (pid 7800)
a.out killed (pid 7801)
a.out killed (pid 7804)
a.out killed (pid 7805)
a.out killed (pid 7806)
a.out killed (pid 7807)
a.out killed (pid 7813)
a.out killed (pid 7814)
a.out killed (pid 7815)
a.out killed (pid 7817)
a.out killed (pid 7818)
a.out killed (pid 7819)
[2]+  Terminated              ./a.out
alexander@alexander-VirtualBox:~$ ps --forest -o user,pgid,ppid,pid,state,cmd
USER        PGID    PPID     PID S CMD
alexand+    1960    1955    1960 S bash
alexand+    7860    1960    7860 R  \_ ps --forest -o user,pgid,ppid,pid,state,cmd
alexander@alexander-VirtualBox:~$ 
```

Htop



```
    0[|||
    1[
    2[|||
    3[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
  Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||||||                               944M
  Swp[                                                                                         0

  Main  I/O
    PID USER      PRI  NI  VIRT   RES   SHR S  CPU% MEM%   TIME+  Command
   2134 alexander  20   0  344M 24808 17736 S   0.0  0.3  0:00.01 /usr/libexec/xdg-desktop-portal
   2136 alexander  20   0  198M 25276 18108 S   0.0  0.3  0:00.21 /usr/libexec/ibus-x11
   2140 alexander  20   0  198M 25276 18108 S   0.0  0.3  0:00.00 /usr/libexec/ibus-x11
   2141 alexander  20   0  198M 25276 18108 S   0.0  0.3  0:00.00 /usr/libexec/ibus-x11
   2142 alexander  20   0  158M  2680  2284 S   0.0  0.0  0:00.00 /usr/bin/VBoxClient --draganddr
   2143 alexander  20   0  158M  2680  2284 S   0.0  0.0  0:00.00 /usr/bin/VBoxClient --draganddr
   2144 alexander  20   0  158M  2680  2284 S   0.0  0.0  0:01.80 /usr/bin/VBoxClient --draganddr
   2145 alexander  20   0  157M  2740  2348 S   0.0  0.0  0:00.00 /usr/bin/VBoxClient --seamless
   2146 alexander  20   0  157M  2740  2348 S   0.0  0.0  0:00.00 /usr/bin/VBoxClient --seamless
   2147 alexander  20   0 5182M  388M  131M S   0.0  4.9  0:00.00 /usr/bin/gnome-shell
   2161 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.94 gjs /usr/share/gnome-shell/exte
   2163 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.00 gjs /usr/share/gnome-shell/exte
   2164 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.08 gjs /usr/share/gnome-shell/exte
   2165 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.01 gjs /usr/share/gnome-shell/exte
   2166 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.00 gjs /usr/share/gnome-shell/exte
   2167 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.00 gjs /usr/share/gnome-shell/exte
   2168 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.00 gjs /usr/share/gnome-shell/exte
   2169 alexander  20   0  617M 14384 11912 S   0.0  0.2  0:00.00 /usr/libexec/xdg-desktop-portal
   2170 alexander  20   0  617M 14384 11912 S   0.0  0.2  0:00.00 /usr/libexec/xdg-desktop-portal
   2172 alexander  20   0 2859M 58288 42212 S   0.0  0.7  0:00.00 gjs /usr/share/gnome-shell/exte
   2237 alexander  20   0 44576 19336 11592 S   0.0  0.2  0:00.04 /usr/bin/python3 /usr/bin/gnome
   2240 alexander  20   0  382M 27508 19968 S   0.0  0.3  0:00.10 /usr/bin/gnome-terminal.real --
   2241 alexander  20   0  382M 27508 19968 S   0.0  0.3  0:00.00 /usr/bin/gnome-terminal.real --
   2243 alexander  20   0  382M 27508 19968 S   0.0  0.3  0:00.00 /usr/bin/gnome-terminal.real --
   2244 alexander  20   0  382M 27508 19968 S   0.0  0.3  0:00.00 /usr/bin/gnome-terminal.real --
   2245 alexander  20   0  556M 59332 45984 S   0.0  0.7  0:07.67 /usr/libexec/gnome-terminal-ser
   2246 alexander  20   0  556M 59332 45984 S   0.0  0.7  0:00.00 /usr/libexec/gnome-terminal-ser
   2248 alexander  20   0  556M 59332 45984 S   0.0  0.7  0:00.00 /usr/libexec/gnome-terminal-ser
   2249 alexander  20   0  556M 59332 45984 S   0.0  0.7  0:00.00 /usr/libexec/gnome-terminal-ser
   2263 alexander  20   0 20192  5880  3972 S   0.0  0.1  0:00.08 bash
   2273 alexander  20   0  490M 29344 21460 S   0.0  0.4  0:00.22 update-notifier
   2278 alexander  20   0  490M 29344 21460 S   0.0  0.4  0:00.02 update-notifier
   2280 alexander  20   0  490M 29344 21460 S   0.0  0.4  0:00.00 update-notifier
   2281 alexander  20   0  490M 29344 21460 S   0.0  0.4  0:00.00 update-notifier
   2470 alexander  20   0 5182M  388M  131M S   0.0  4.9  0:00.00 /usr/bin/gnome-shell
   2574 alexander  30  10  665M  159M 99984 S   0.0  2.0  0:05.39 /usr/bin/python3 /usr/bin/updat
   2575 alexander  30  10  665M  159M 99984 S   0.0  2.0  0:00.00 /usr/bin/python3 /usr/bin/updat
   2577 alexander  30  10  665M  159M 99984 S   0.0  2.0  0:00.00 /usr/bin/python3 /usr/bin/updat
   2578 alexander  30  10  665M  159M 99984 S   0.0  2.0  0:00.00 /usr/bin/python3 /usr/bin/updat
   3483 alexander  20   0  779M 27688 21372 S   0.0  0.3  0:00.00 /usr/libexec/gsd-power
   3538 alexander  20   0  2640   944   856 R 112.0  0.0  1:29.81 ./a.out
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

q -- fast exit

/ -- search by process name

Search engines

Regular expressions (`man grep` + `/BRE` + Enter)

Google regex101.com for training and testing regular expressions.

Regexepr support BRE (basic), ERE (extended) and PCRE (Perl) syntax. BRE syntax is generally the same in different utilities, but ERE and PCRE can be a little bit different in some utilities as `awk`, `sed`, `grep`, `python3`, etc.

Idea (step-by-step thinking)

1. Target characters:

   ```
   a b .. z A .. Z
   0 1 2 .. 9
   \( \" \t \n \' \{
   ```

   Additional special symbols: line start, end, word boundary:

   ```
   ^ $ \b
   ```

   or ANY SYMBOL:

```
.
```

2. Grouping and groups (unique for every utility or engine):

```
# general sequence of characters group
abc
# "one of us" group
[abc]
# "all digits" group - see grep manual
[::digit::]
```

3. Group repetition symbols (applies to groups):

   Zero or more / one or more / zero or one / non-matched group / repetition count

```
* + ? [^abc] {2,4}
```

4. Back references (not everywhere) - refer to previous matched groups:

```
\1 \2 \3
```

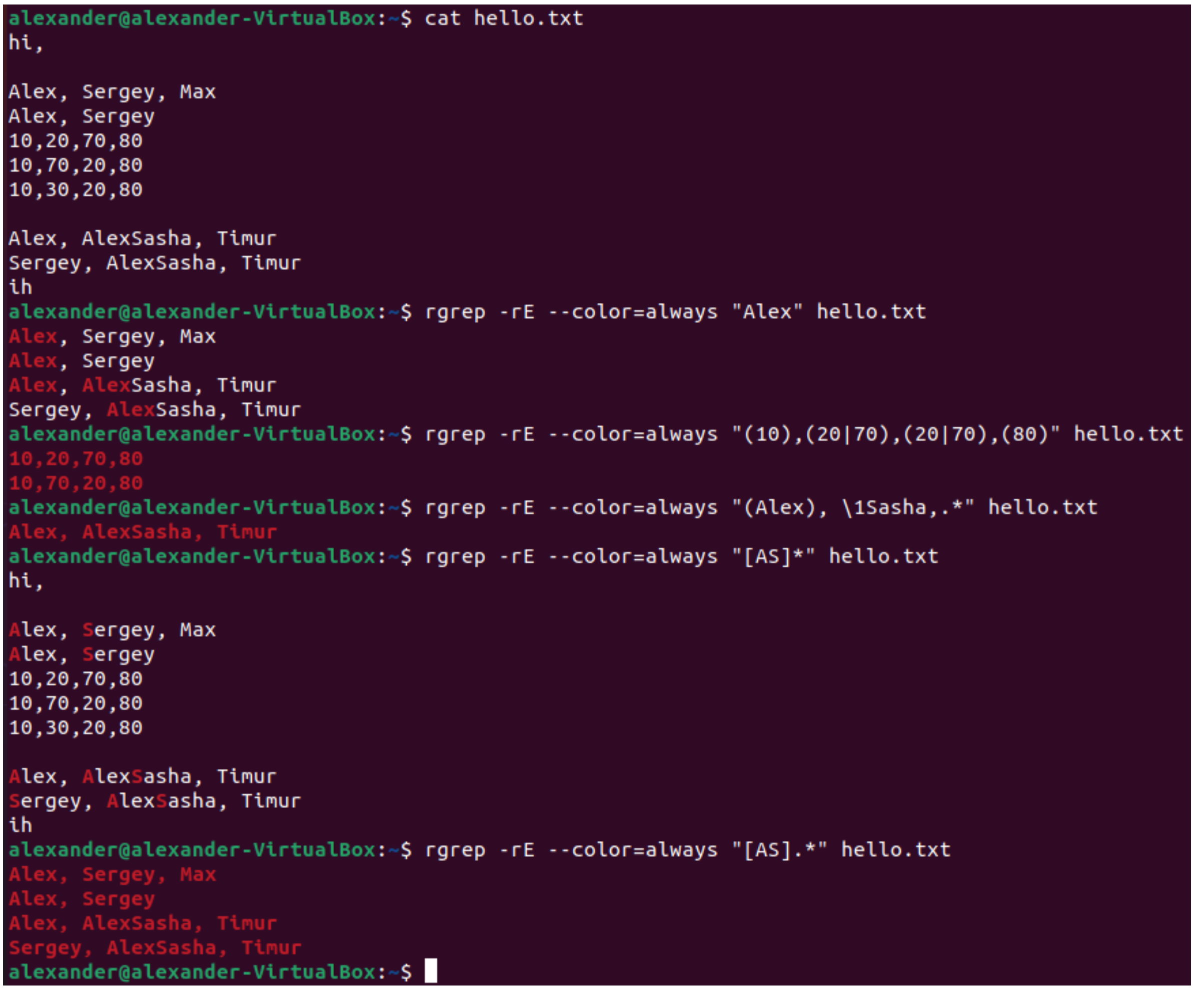
> Number means position number of matched group, group should surrounded by braces `( )`.

5. Flags (global text search, or line by line search, or smth else)

```
\g \w ...
```

> Advice: use `rgrep`, not `grep`. It works faster!

Examples



```
alexander@alexander-VirtualBox:~$ cat hello.txt
hi,

Alex, Sergey, Max
Alex, Sergey
10,20,70,80
10,70,20,80
10,30,20,80

Alex, AlexSasha, Timur
Sergey, AlexSasha, Timur
ih
alexander@alexander-VirtualBox:~$ rgrep -rE --color=always "Alex" hello.txt
Alex, Sergey, Max
Alex, Sergey
Alex, AlexSasha, Timur
Sergey, AlexSasha, Timur
alexander@alexander-VirtualBox:~$ rgrep -rE --color=always "(10),(20|70),(20|70),(80)" hello.txt
10,20,70,80
10,70,20,80
alexander@alexander-VirtualBox:~$ rgrep -rE --color=always "(Alex), \1Sasha,.*" hello.txt
Alex, AlexSasha, Timur
alexander@alexander-VirtualBox:~$ rgrep -rE --color=always "[AS]*" hello.txt
hi,

Alex, Sergey, Max
Alex, Sergey
10,20,70,80
10,70,20,80
10,30,20,80

Alex, AlexSasha, Timur
Sergey, AlexSasha, Timur
ih
alexander@alexander-VirtualBox:~$ rgrep -rE --color=always "[AS].*" hello.txt
Alex, Sergey, Max
Alex, Sergey
Alex, AlexSasha, Timur
Sergey, AlexSasha, Timur
alexander@alexander-VirtualBox:~$ 
```

Wildcards

Do not mix wildcard and regex patterns! They use some same symbols as `?`, `*`, `[-]` but have DIFFERENT INTERPRETATION!

Wildcard meaning

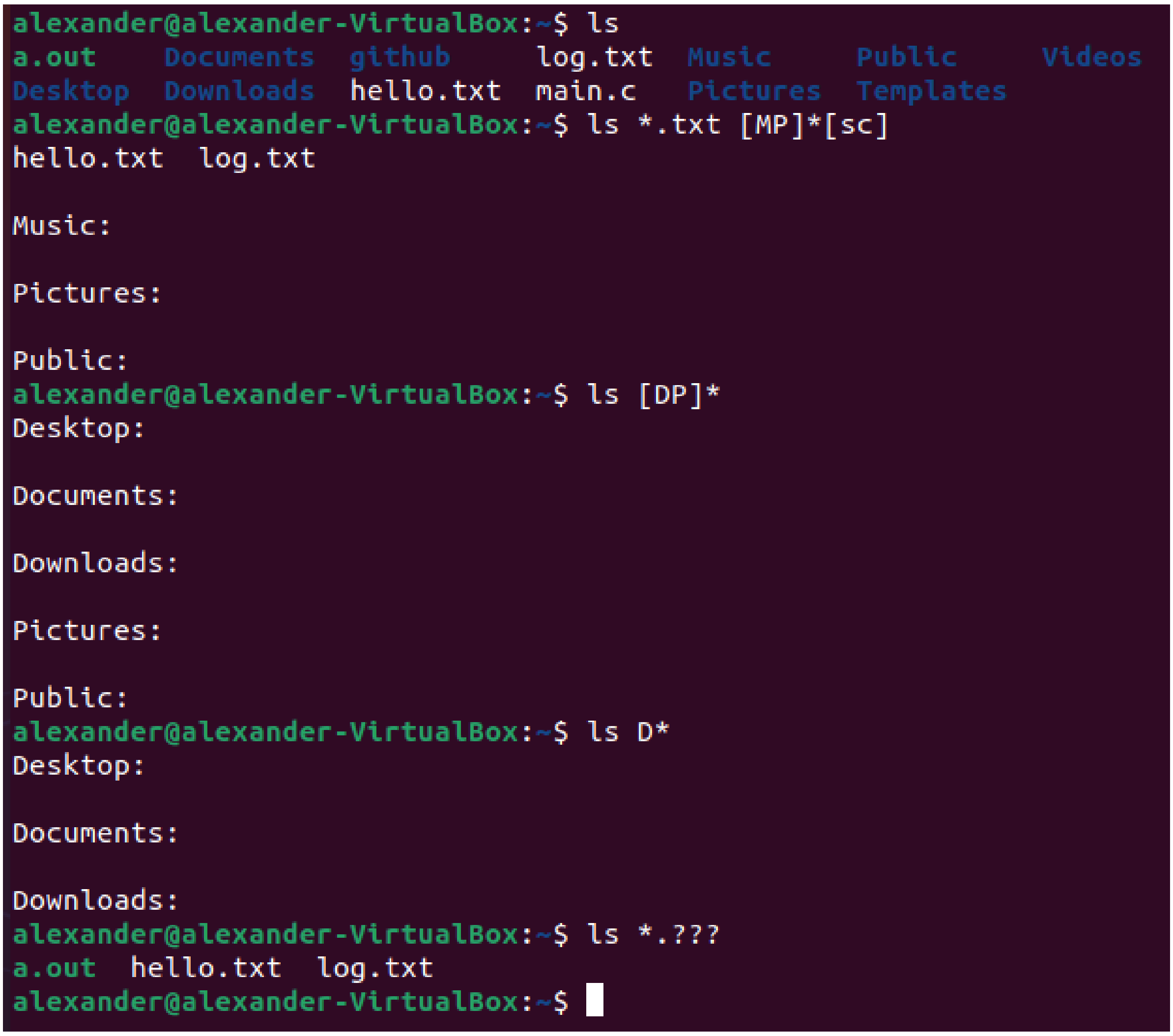
`*` - any sequence of symbols (not applied as group repetition symbol)

> Note:
>
> Wildcard `*` **does not match hidden files**.

`?` - zero or one symbol ((not applied as group repetition symbol))

`[abc]`, `[a-c]` - group of ONE character that can be `a`, `b` or `c`.

Wildcards usage examples



```
alexander@alexander-VirtualBox:~$ ls
a.out     Documents  github     log.txt  Music      Public     Videos
Desktop   Downloads  hello.txt  main.c   Pictures   Templates
alexander@alexander-VirtualBox:~$ ls *.txt [MP]*[sc]
hello.txt  log.txt

Music:

Pictures:

Public:
alexander@alexander-VirtualBox:~$ ls [DP]*
Desktop:

Documents:

Downloads:

Pictures:

Public:
alexander@alexander-VirtualBox:~$ ls D*
Desktop:

Documents:

Downloads:
alexander@alexander-VirtualBox:~$ ls *.???
a.out  hello.txt  log.txt
alexander@alexander-VirtualBox:~$ 
```

Inline array syntax

```
list=(sasha masha sasa a h)
# print num of elems
echo ${#list[*]}

# print elem indexes
for i in ${!list[*]}; do echo $i; done

# print strlen() of one's elem
echo ${#list[0]}

# print elems in idx range [2:5]
echo ${list[*]:2:5}

# print all w/ exclude
for i in ${list[@]}; do if ! [[ $i =~ ^(sasha|masha) ]]; then echo $i; fi; done
for i in ${list[@]}; do if ! [[ $i =~ ^(s) ]]; then echo $i; fi; done
for i in ${list[@]}; do if ! [[ $i =~ ^(s$) ]]; then echo $i; fi; done
```

VIM (`sudo apt-get install vim`)

Vim Basics

Insert / ESC - set insert / visual mode for enable / disable editing

`:q` - exit vim

When visual mode enabled, you can do the following:

`*` - find a match pointed by cursor

`:n` go to line n

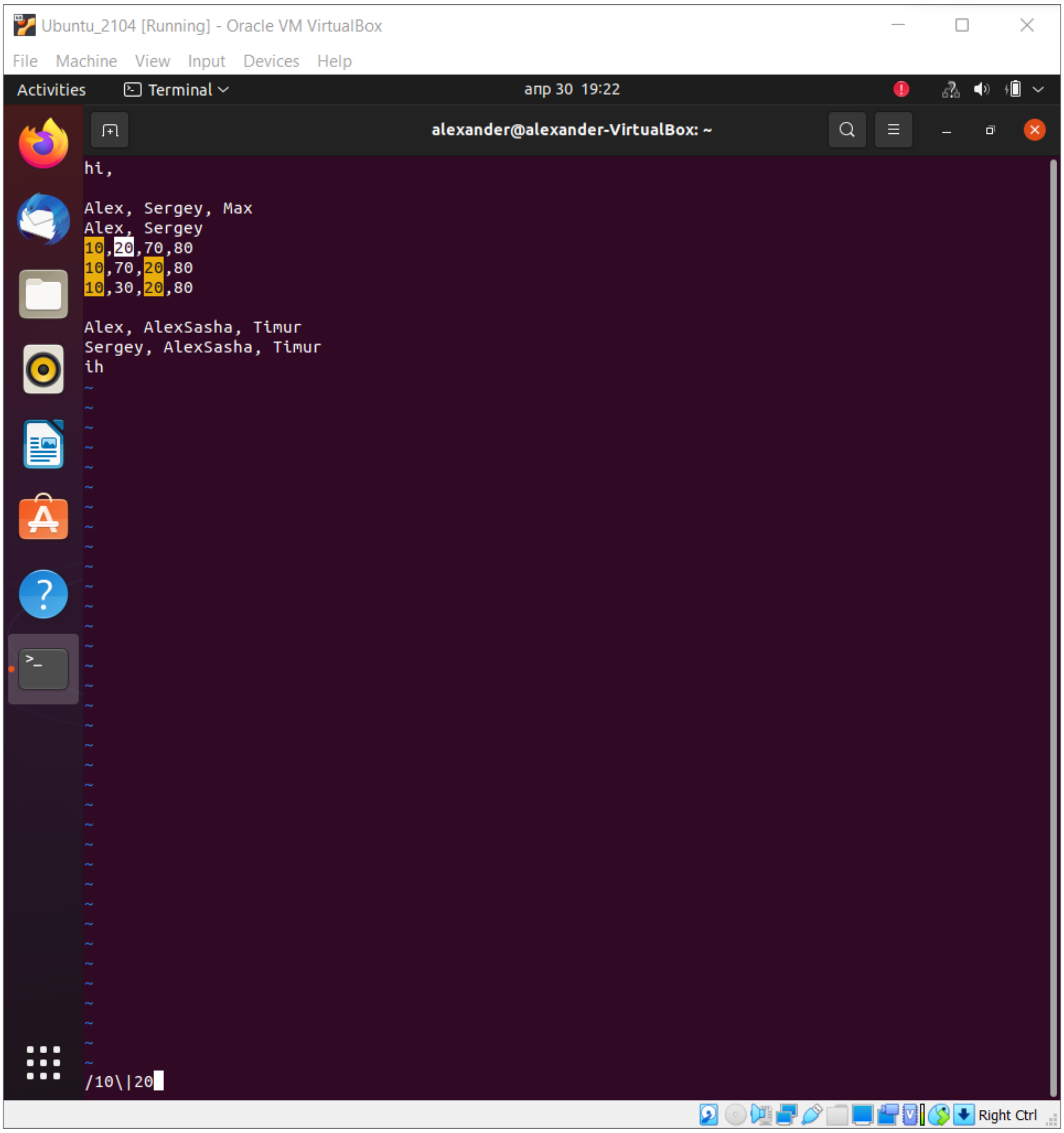
`:1` or `:^` - go to start (like **g** in `less`)

`:1000000000....` - go to end (like **Shift+g** in `less`)

`:$` - go to end

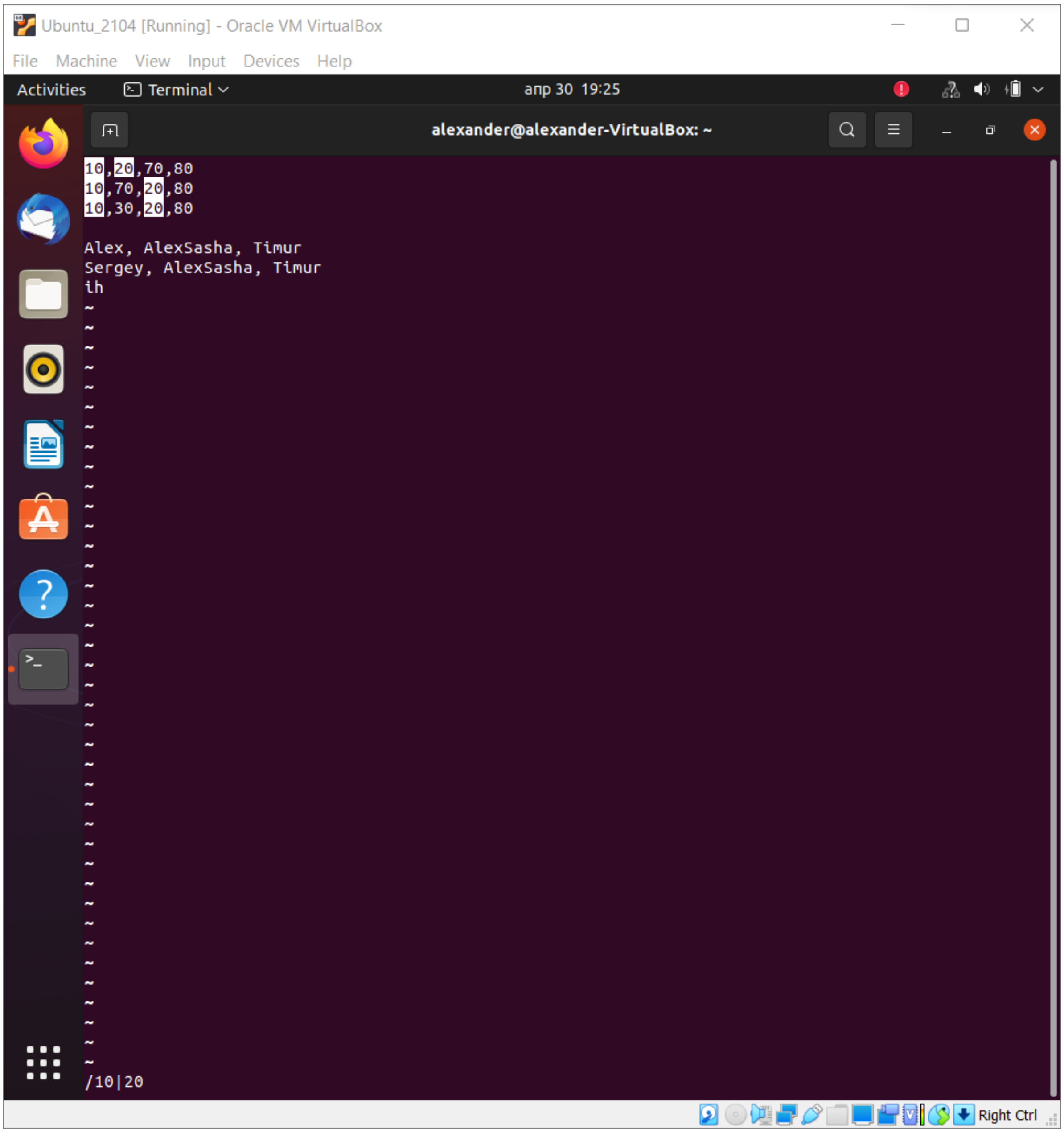
Ctrl+g - view name of file

`/<keyword>` - search pattern


Ubuntu_2104 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Activities
Terminal
апр 30 19:22
alexander@alexander-VirtualBox: ~
Right Ctrl

```
hi,

Alex, Sergey, Max
Alex, Sergey
10,20,70,80
10,70,20,80
10,30,20,80

Alex, AlexSasha, Timur
Sergey, AlexSasha, Timur
ih
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
/10\|20
```

> Enter command `:set hlsearch` for yallow highlight of all patterns. Pay attention to regex pipe: `\|`. In less the command is differnt a bit.


Ubuntu_2104 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
апр 30 19:25
alexander@alexander-VirtualBox: ~
Right Ctrl

```
10,20,70,80
10,70,20,80
10,30,20,80

Alex, AlexSasha, Timur
Sergey, AlexSasha, Timur
ih
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
/10|20
```

You can configure Vim editor by adding new file `~/.vimrc` (or it has existed already) and add:

```
set mouse=a             "" mouse activation
set number              "" numeric lines
syntax on               "" highlight syntax
set listchars=space:.   "" type :set list / :set nolist for marking/unmarking spaces in a code

vmap <C-c> hyi          ""
vmap <C-v> hdi
imap <C-u> <ESC>ui

set nowrap
set rnet
set hlsearch
set incsearch
command Wq execute "wq"
set smartindent
set tabstop=4
set shiftwidth=4
```

Open multiple files

```
vim -O file1.txt file2.txt
```

Switch between windows:

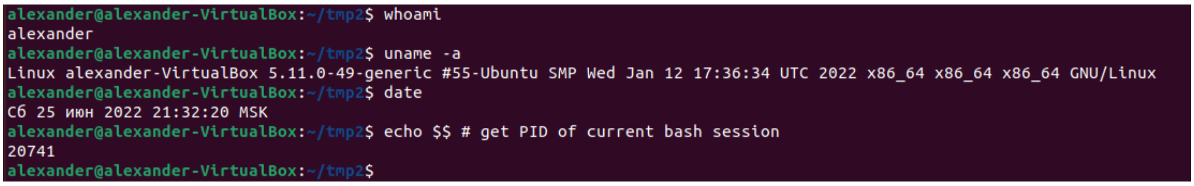
Ctrl+W W

System info

OS basics (if you got "*knocked out!*")

```
alexander@alexander-VirtualBox:~/tmp2$ whatis --section=1 whoami uname date
whoami (1)           - print effective userid
uname (1)            - print system information
date (1)             - print or set the system date and time
```

```
# get pid of current terminal session
# can be used for learing linux process (ls /procs/$$/*)
echo $$
```



```
alexander@alexander-VirtualBox:~/tmp2$ whoami
alexander
alexander@alexander-VirtualBox:~/tmp2$ uname -a
Linux alexander-VirtualBox 5.11.0-49-generic #55-Ubuntu SMP Wed Jan 12 17:36:34 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
alexander@alexander-VirtualBox:~/tmp2$ date
Сб 25 июн 2022 21:32:20 MSK
alexander@alexander-VirtualBox:~/tmp2$ echo $$ # get PID of current bash session
20741
alexander@alexander-VirtualBox:~/tmp2$
```

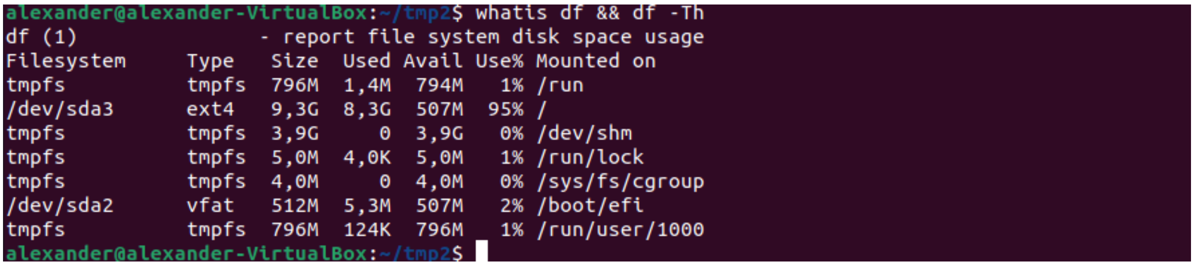
Memory usage

RAM

```
# info
alexander@alexander-VirtualBox:~/tmp2$ whatis --section=1 free
free (1)             - Display amount of free and used memory in the system
alexander@alexander-VirtualBox:~/tmp2$ free
              total        used        free      shared  buff/cache   available
Mem:        8143196      665640     5104104       28156     2373452     7198696
Swap:        459208           0      459208
```

Disks & filesystems (see src)

Global



```
alexander@alexander-VirtualBox:~/tmp2$ whatis df && df -Th
df (1)               - report file system disk space usage
Filesystem     Type   Size  Used Avail Use% Mounted on
tmpfs          tmpfs  796M  1,4M  794M   1% /run
/dev/sda3      ext4   9,3G  8,3G  507M  95% /
tmpfs          tmpfs  3,9G     0  3,9G   0% /dev/shm
tmpfs          tmpfs  5,0M  4,0K  5,0M   1% /run/lock
tmpfs          tmpfs  4,0M     0  4,0M   0% /sys/fs/cgroup
/dev/sda2      vfat   512M  5,3M  507M   2% /boot/efi
tmpfs          tmpfs  796M  124K  796M   1% /run/user/1000
alexander@alexander-VirtualBox:~/tmp2$
```

Current directory

```
# info
alexander@alexander-VirtualBox:/home$ whatis du
du (1)               - estimate file space usage

# example 1
alexander@alexander-VirtualBox:/home$ du -h --max-depth=1 ~
3,0M    /home/alexander/snap
236K    /home/alexander/Pictures
30M /home/alexander/.mozilla
4,0K    /home/alexander/Music
4,0K    /home/alexander/Templates
4,0K    /home/alexander/tmp2
4,0K    /home/alexander/Public
196K    /home/alexander/.config
2,4M    /home/alexander/github
16K /home/alexander/.ssh
4,0K    /home/alexander/Documents
4,0K    /home/alexander/Desktop
104M    /home/alexander/.cache
4,0K    /home/alexander/Downloads
792K    /home/alexander/.local
4,0K    /home/alexander/Videos
12K /home/alexander/.dbus
141M    /home/alexander

# example 2
alexander@alexander-VirtualBox:/home$ du -sh ~
141M    /home/alexander

# example 3
alexander@alexander-VirtualBox:/home$ du -sb ~
143074275   /home/alexander

# example 4
alexander@alexander-VirtualBox:/home$ du -sm ~
141 /home/alexander
```

CPU

> CPU - state machine with **combinational** and **sequential** logic.

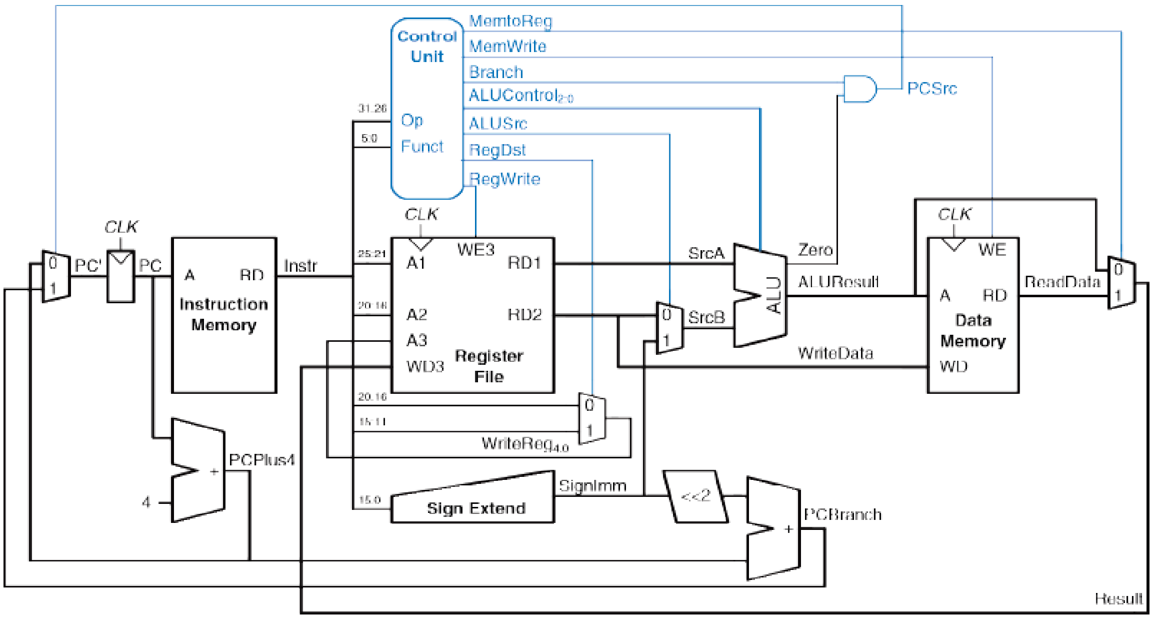


Рис. 7.11 Законченный однотактный процессор MIPS

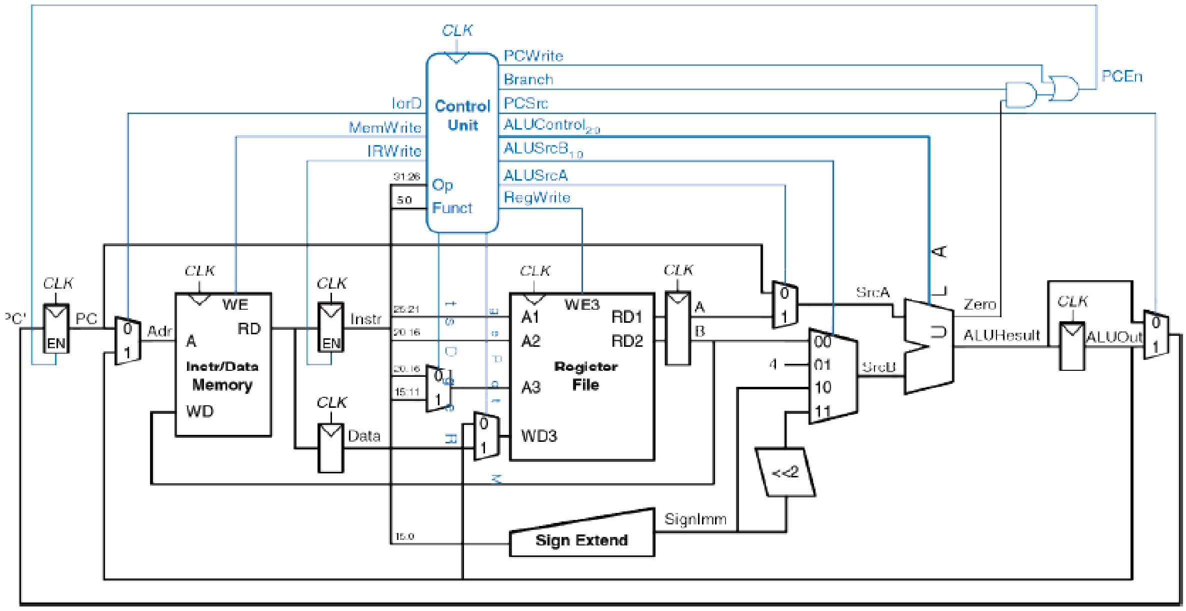


Рис. 7.27 Законченный многотактный процессор MIPS

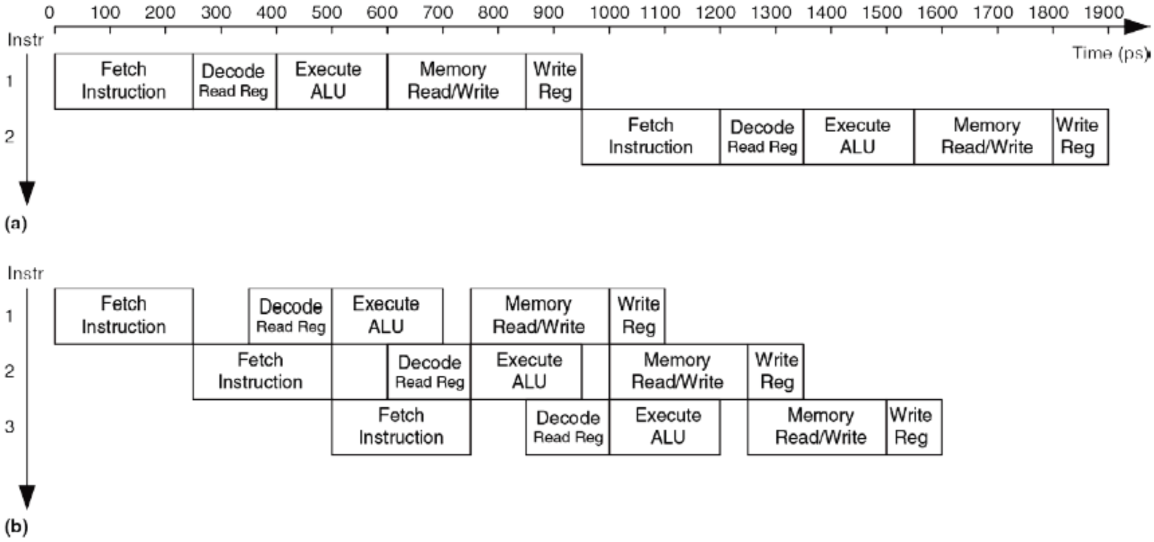


Рис. 7.43 Временная диаграмма однотактного (a) и конвейерного (b) процессоров

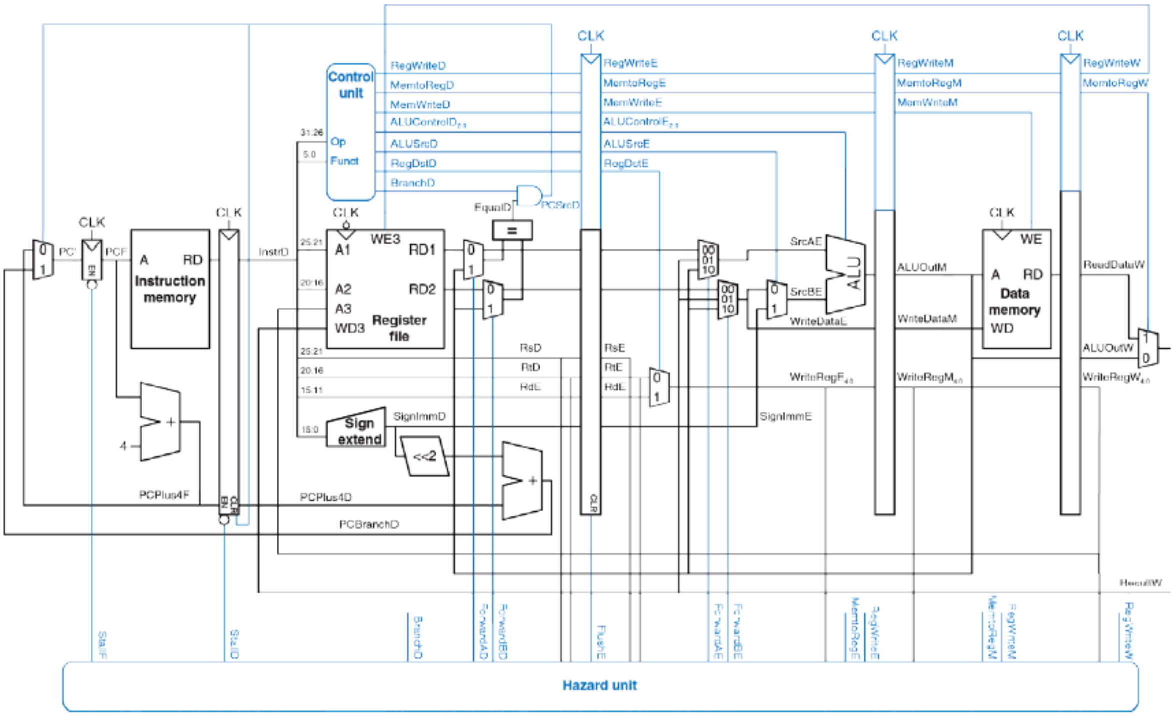


Рис. 7.58 Конвейерный процессор, разрешающий все конфликты

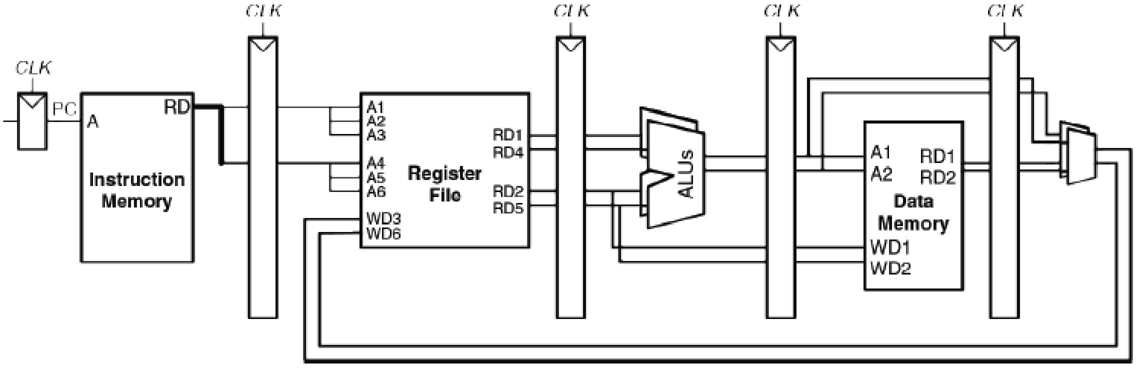
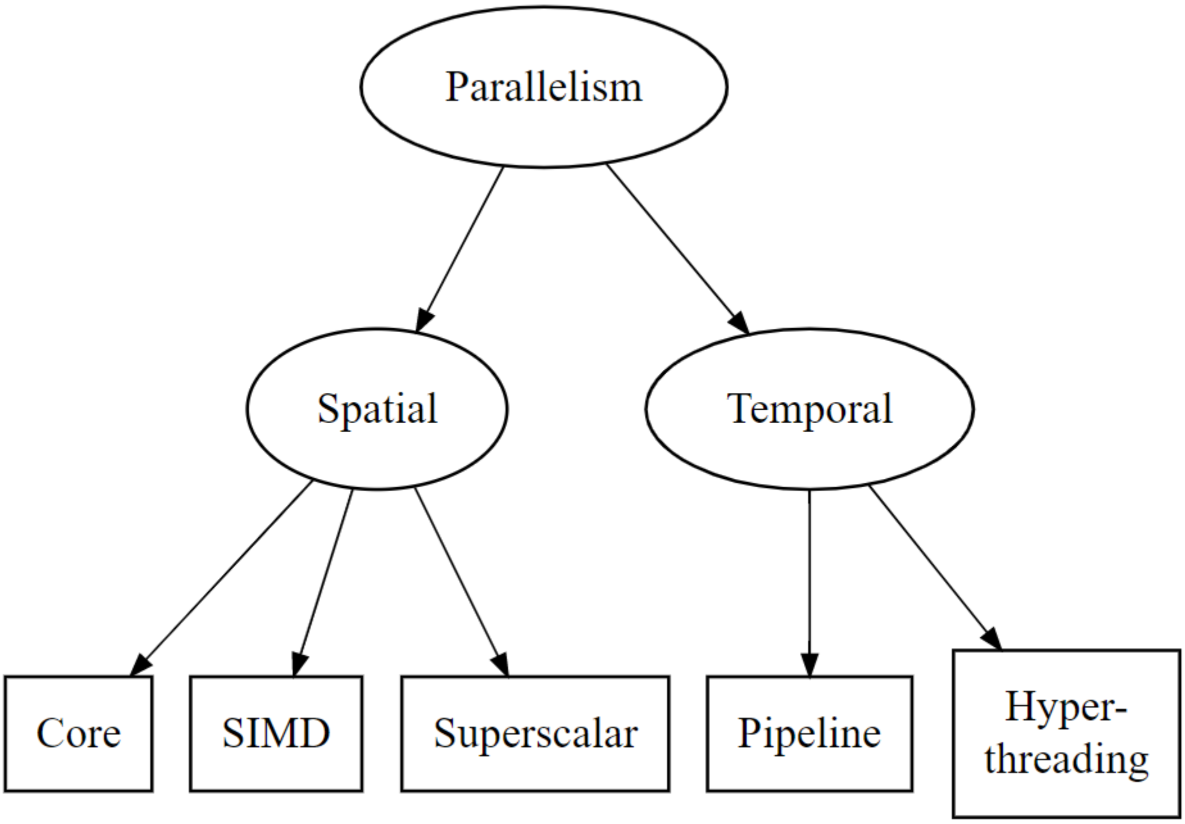
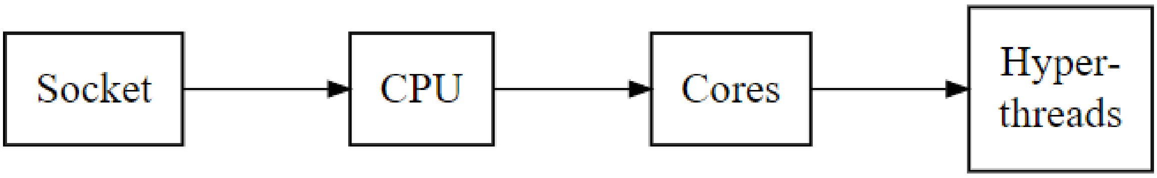


Рис. 7.67 Тракт данных суперскалярного процессора

Types of parallelism

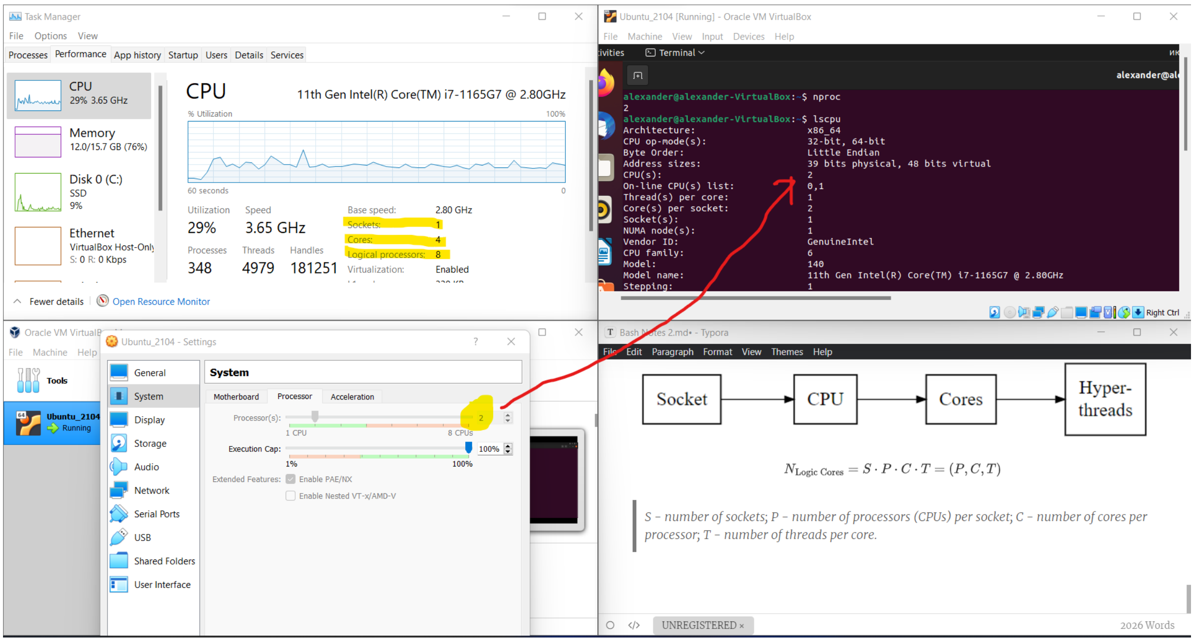


Processing hierarchy (architecture topology)



$$N_{\text{Logic Cores}} = S \cdot P \cdot C \cdot T = (P, C, T) \tag{2}$$

> S - number of sockets; P - number of processors (CPUs) per socket; C - number of cores per processor; T - number of threads per core.



CPU 29% 3.65 GHz

Memory 12.0/15.7 GB (76%)

Disk 0 (C:) SSD 9%

Ethernet VirtualBox Host-Only S: 0 R: 0 Kbps

CPU 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz

% Utilization 100%

60 seconds 0

Utilization	Speed		Base speed:	2.80 GHz
29%	3.65 GHz		Sockets:	1
			Cores:	4
Processes	Threads	Handles	Logical processors:	8
348	4979	181251	Virtualization:	Enabled

System

Processor(s): 2 (1 CPU – 8 CPUs)

Execution Cap: 100% (1% – 100%)

Extended Features: ☑ Enable PAE/NX

☐ Enable Nested VT-x/AMD-V

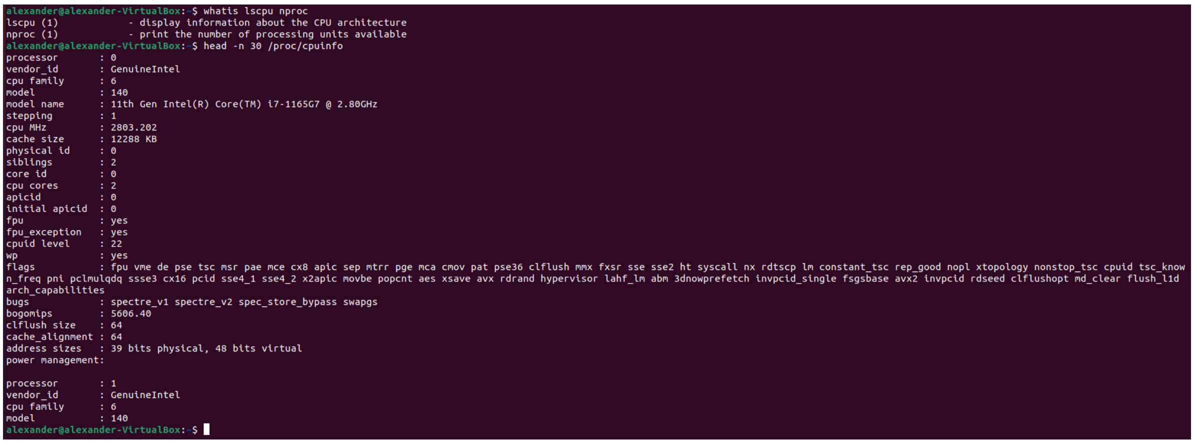
```
alexander@alexander-VirtualBox:~$ nproc
2
alexander@alexander-VirtualBox:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   39 bits physical, 48 bits virtual
CPU(s):                          2
On-line CPU(s) list:             0,1
Thread(s) per core:              1
Core(s) per socket:              2
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           140
Model name:                      11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
Stepping:                        1
```

Socket → CPU → Cores → Hyper-threads

$N_{\text{Logic Cores}} = S \cdot P \cdot C \cdot T = (P, C, T)$

S - number of sockets; P - number of processors (CPUs) per socket; C - number of cores per processor; T - number of threads per core.

Statistics



```
alexander@alexander-VirtualBox:~$ whatis lscpu nproc
lscpu (1)            - display information about the CPU architecture
nproc (1)            - print the number of processing units available
alexander@alexander-VirtualBox:~$ head -n 30 /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 140
model name      : 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
stepping        : 1
cpu MHz         : 2803.202
cache size      : 12288 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_know
n_freq pni pclmulqdq ssse3 cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single fsgsbase avx2 invpcid rdseed clflushopt md_clear flush_l1d
arch_capabilities
bugs            : spectre_v1 spectre_v2 spec_store_bypass swapgs
bogomips        : 5606.40
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 6
model           : 140
alexander@alexander-VirtualBox:~$ 
```

Kernel configuration

> Upd: о виртуальных файловых системах
>
> - /sys - интерфейс для конфигурации ядра в реальном времени + девайсов
> - /proc - интерфейс для управления процессами

```
alexander@alexander-VirtualBox:~$ whatis sysctl
sysctl (8)           - configure kernel parameters at runtime
```

> `sysctl` configuration changes are **active before reboot.**

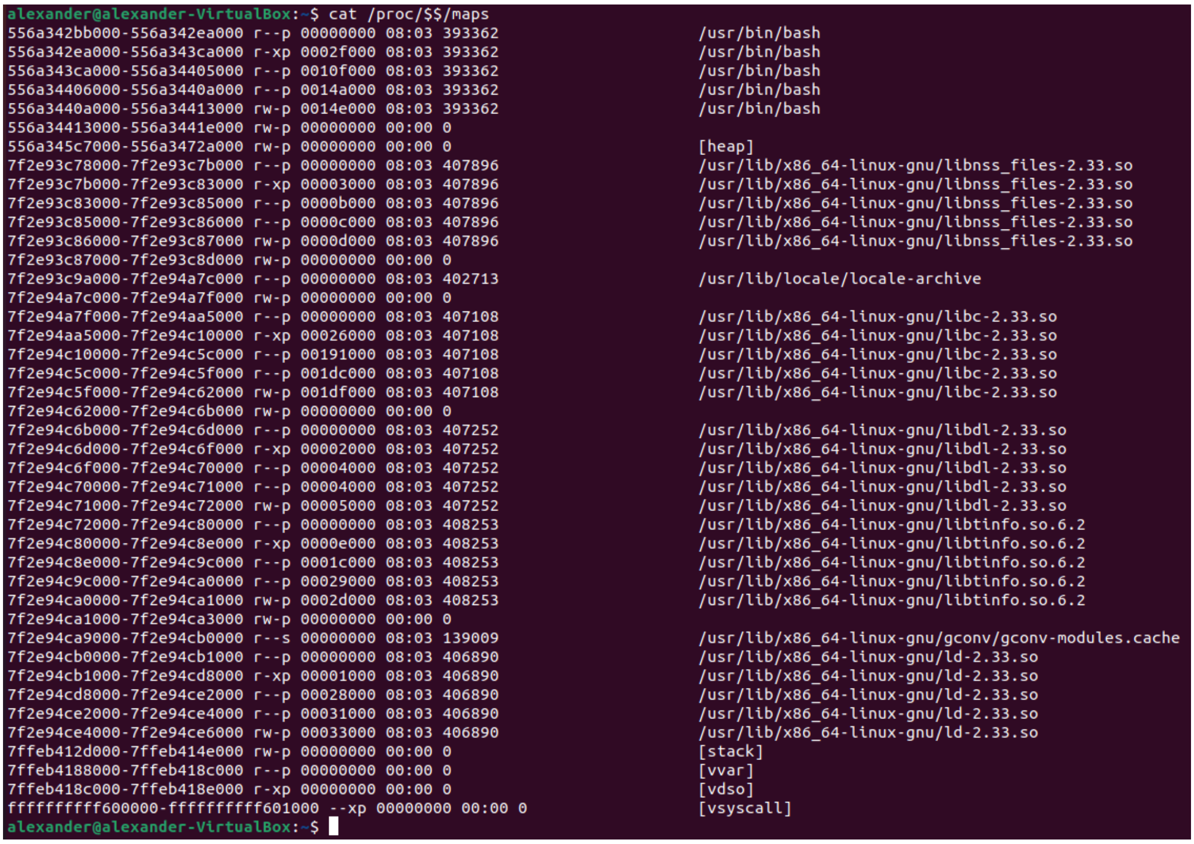
```
# show kernel configuration
sysctl -a | less -S
# edit kernel configuration
alexander@alexander-VirtualBox:~$ sudo sysctl kernel.hostname=alectroid
# or
sudo echo alectroid > /proc/sys/kernel/hostname

# Ctrl + Alt + T
alexander@alectroid:~$

# load settings
sudo sysctl --load # load from /etc/sysctl.conf
sudo sysctl --load <filname>.conf # load from user's file w/ .conf suffix
sudo sysctl --system # load from all kernel config file in the following priority

```

Process information

```
proc (5)             - process information pseudo-filesystem
```

Contains files & directories belonged to all system of processes. Directories `/proc/<pid>/` contain the whole information about process with current `<pid>`. For exapmle:

```
# display mappings of current process
cat /proc/$$/maps
```



```
alexander@alexander-VirtualBox:~$ cat /proc/$$/maps
556a342bb000-556a342ea000 r--p 00000000 08:03 393362                     /usr/bin/bash
556a342ea000-556a343ca000 r-xp 0002f000 08:03 393362                     /usr/bin/bash
556a343ca000-556a34405000 r--p 0010f000 08:03 393362                     /usr/bin/bash
556a34406000-556a3440a000 r--p 0014a000 08:03 393362                     /usr/bin/bash
556a3440a000-556a34413000 rw-p 0014e000 08:03 393362                     /usr/bin/bash
556a34413000-556a3441e000 rw-p 00000000 00:00 0
556a345c7000-556a3472a000 rw-p 00000000 00:00 0                          [heap]
7f2e93c78000-7f2e93c7b000 r--p 00000000 08:03 407896                     /usr/lib/x86_64-linux-gnu/libnss_files-2.33.so
7f2e93c7b000-7f2e93c83000 r-xp 00003000 08:03 407896                     /usr/lib/x86_64-linux-gnu/libnss_files-2.33.so
7f2e93c83000-7f2e93c85000 r--p 0000b000 08:03 407896                     /usr/lib/x86_64-linux-gnu/libnss_files-2.33.so
7f2e93c85000-7f2e93c86000 r--p 0000c000 08:03 407896                     /usr/lib/x86_64-linux-gnu/libnss_files-2.33.so
7f2e93c86000-7f2e93c87000 rw-p 0000d000 08:03 407896                     /usr/lib/x86_64-linux-gnu/libnss_files-2.33.so
7f2e93c87000-7f2e93c8d000 rw-p 00000000 00:00 0
7f2e93c9a000-7f2e94a7c000 r--p 00000000 08:03 402713                     /usr/lib/locale/locale-archive
7f2e94a7c000-7f2e94a7f000 rw-p 00000000 00:00 0
7f2e94a7f000-7f2e94aa5000 r--p 00000000 08:03 407108                     /usr/lib/x86_64-linux-gnu/libc-2.33.so
7f2e94aa5000-7f2e94c10000 r-xp 00026000 08:03 407108                     /usr/lib/x86_64-linux-gnu/libc-2.33.so
7f2e94c10000-7f2e94c5c000 r--p 00191000 08:03 407108                     /usr/lib/x86_64-linux-gnu/libc-2.33.so
7f2e94c5c000-7f2e94c5f000 r--p 001dc000 08:03 407108                     /usr/lib/x86_64-linux-gnu/libc-2.33.so
7f2e94c5f000-7f2e94c62000 rw-p 001df000 08:03 407108                     /usr/lib/x86_64-linux-gnu/libc-2.33.so
7f2e94c62000-7f2e94c6b000 rw-p 00000000 00:00 0
7f2e94c6b000-7f2e94c6d000 r--p 00000000 08:03 407252                     /usr/lib/x86_64-linux-gnu/libdl-2.33.so
7f2e94c6d000-7f2e94c6f000 r-xp 00002000 08:03 407252                     /usr/lib/x86_64-linux-gnu/libdl-2.33.so
7f2e94c6f000-7f2e94c70000 r--p 00004000 08:03 407252                     /usr/lib/x86_64-linux-gnu/libdl-2.33.so
7f2e94c70000-7f2e94c71000 r--p 00004000 08:03 407252                     /usr/lib/x86_64-linux-gnu/libdl-2.33.so
7f2e94c71000-7f2e94c72000 rw-p 00005000 08:03 407252                     /usr/lib/x86_64-linux-gnu/libdl-2.33.so
7f2e94c72000-7f2e94c80000 r--p 00000000 08:03 408253                     /usr/lib/x86_64-linux-gnu/libtinfo.so.6.2
7f2e94c80000-7f2e94c8e000 r-xp 0000e000 08:03 408253                     /usr/lib/x86_64-linux-gnu/libtinfo.so.6.2
7f2e94c8e000-7f2e94c9c000 r--p 0001c000 08:03 408253                     /usr/lib/x86_64-linux-gnu/libtinfo.so.6.2
7f2e94c9c000-7f2e94ca0000 r--p 00029000 08:03 408253                     /usr/lib/x86_64-linux-gnu/libtinfo.so.6.2
7f2e94ca0000-7f2e94ca1000 rw-p 0002d000 08:03 408253                     /usr/lib/x86_64-linux-gnu/libtinfo.so.6.2
7f2e94ca1000-7f2e94ca3000 rw-p 00000000 00:00 0
7f2e94ca9000-7f2e94cb0000 r--s 00000000 08:03 139009                     /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
7f2e94cb0000-7f2e94cb1000 r--p 00000000 08:03 406890                     /usr/lib/x86_64-linux-gnu/ld-2.33.so
7f2e94cb1000-7f2e94cd8000 r-xp 00001000 08:03 406890                     /usr/lib/x86_64-linux-gnu/ld-2.33.so
7f2e94cd8000-7f2e94ce2000 r--p 00028000 08:03 406890                     /usr/lib/x86_64-linux-gnu/ld-2.33.so
7f2e94ce2000-7f2e94ce4000 r--p 00031000 08:03 406890                     /usr/lib/x86_64-linux-gnu/ld-2.33.so
7f2e94ce4000-7f2e94ce6000 rw-p 00033000 08:03 406890                     /usr/lib/x86_64-linux-gnu/ld-2.33.so
7ffeb412d000-7ffeb414e000 rw-p 00000000 00:00 0                          [stack]
7ffeb4188000-7ffeb418c000 r--p 00000000 00:00 0                          [vvar]
7ffeb418c000-7ffeb418e000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0                  [vsyscall]
alexander@alexander-VirtualBox:~$
```

process

proc

`man proc, man mmap, man smap`

Литература

- Подборка
- *Роберт Лав* Ядро Линукс. Описание процесса разработки