Universiteit Antwerpen
Faculteit Toegepaste Ingenieurswetenschappen

Lab of 3-Network Architecture

Ruben Nietvelt, Nabeel Nisar Bhat

2024-2025

Scheduled labs for PR01

Session	Date	Subject	Evaluation	Deadline (23:59)
1	01/10/2024	Introduction to the Linux Operating System: Using the shell & exploring the filesystem	Report	07/10/2024
2	08/10/2024	Working with text files and managing running processes	Report	14/10/2024
3	15/10/2024	Writing shell scripts	Report	22/10/2024
4	23/10/2024	Learning system administration, getting & managing software	Report	28/10/2024
5	29/10/2024	Wireshark introduction	Report	05/11/2024
6	06/11/2024	Protocols in action: TCP and UDP	Report	11/11/2024
7	12/11/2024	Ethernet and ARP	Report	19/11/2024
8	20/11/2024	Setting up a DHCP server	Report	25/11/2024
9	26/11/2024	Setting up a DNS server	Report	03/12/2024
10	04/12/2024	Network Address Translation	Report	09/12/2024
11	10/12/2024	Remote Access & Firewalls (1)		**N/A**
12	18/12/2024	Remote Access & Firewalls (2)	Blackboard test	

Scheduled labs for PR02

Session	Date	Subject	Evaluation	Deadline (23:59)
1	02/10/2024	Introduction to the Linux Operating System: Using the shell & exploring the filesystem	Report	08/10/2024
2	09/10/2024	Working with text files and managing running processes	Report	15/10/2024
3	16/10/2024	Writing shell scripts	Report	22/10/2024
4	23/10/2024	Learning system administration, getting & managing software	Report	29/10/2024
5	30/10/2024	Wireshark introduction	Report	05/11/2024
6	06/11/2024	Protocols in action: TCP and UDP	Report	12/11/2024
7	13/11/2024	Ethernet and ARP	Report	19/11/2024
8	20/11/2024	Setting up a DHCP server	Report	26/11/2024
9	27/11/2024	Setting up a DNS server	Report	03/12/2024
10	04/12/2024	Network Address Translation	Report	10/12/2024
11	11/12/2024	Remote Access & Firewalls (1)		**N/A**
12	18/12/2024	Remote Access & Firewalls (2)	Blackboard test	

Important commands: recap

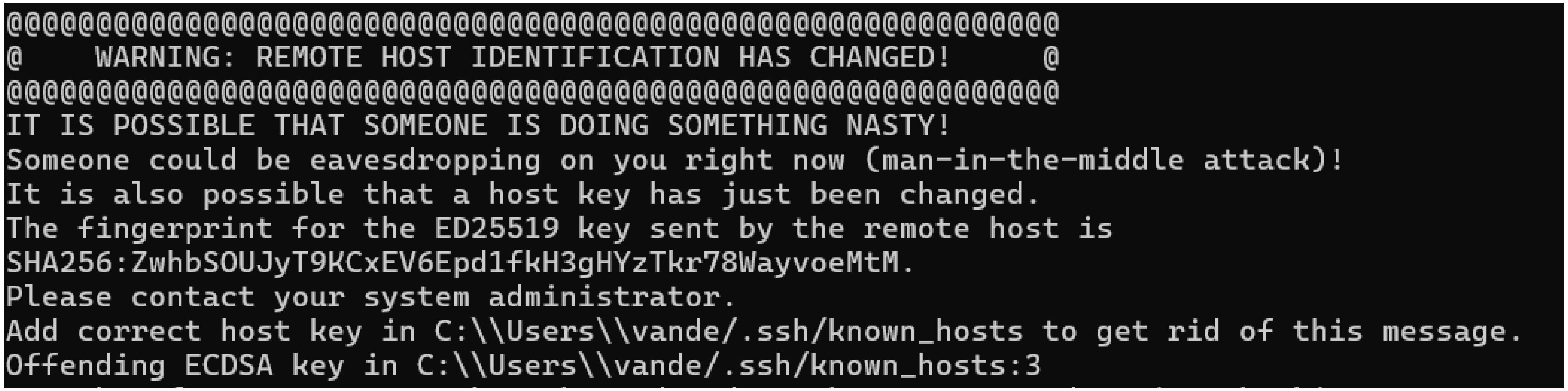
Most important commands

Command	Explanation
ls	Lists directory contents of files and directories.
cd	Change directory.
touch	Updates the access and modification times of each file.
mkdir	Make directory.
cat	Copy content of a file (to terminal or other file).
echo	Display lines of text or strings that are passed as arguments.
sudo	Execute command using root privileges. This access is password-protected and only valid for a limited time. Use ‘sudo su’ for $ ➡ #
apropos	Searching for commands without knowing their exact names.

What went wrong?

What does this mean?

- You are trying to connect via SSH to your VM and are seeing this:



```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:ZwhbSOUJyT9KCxEV6Epd1fkH3gHYzTkr78WayvoeMtM.
Please contact your system administrator.
Add correct host key in C:\\Users\\vande/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in C:\\Users\\vande/.ssh/known_hosts:3
```

Session 2

Working with text files and managing processes

Working with files

Know your editors

While some editors cover the basics, others are merely limited by your imagination...

User-friendly text editors:

- nano
- jed

High input, high return editors:

- vim (vi improved)
- emacs (editor macros)

Know your editors

^ = Ctrl

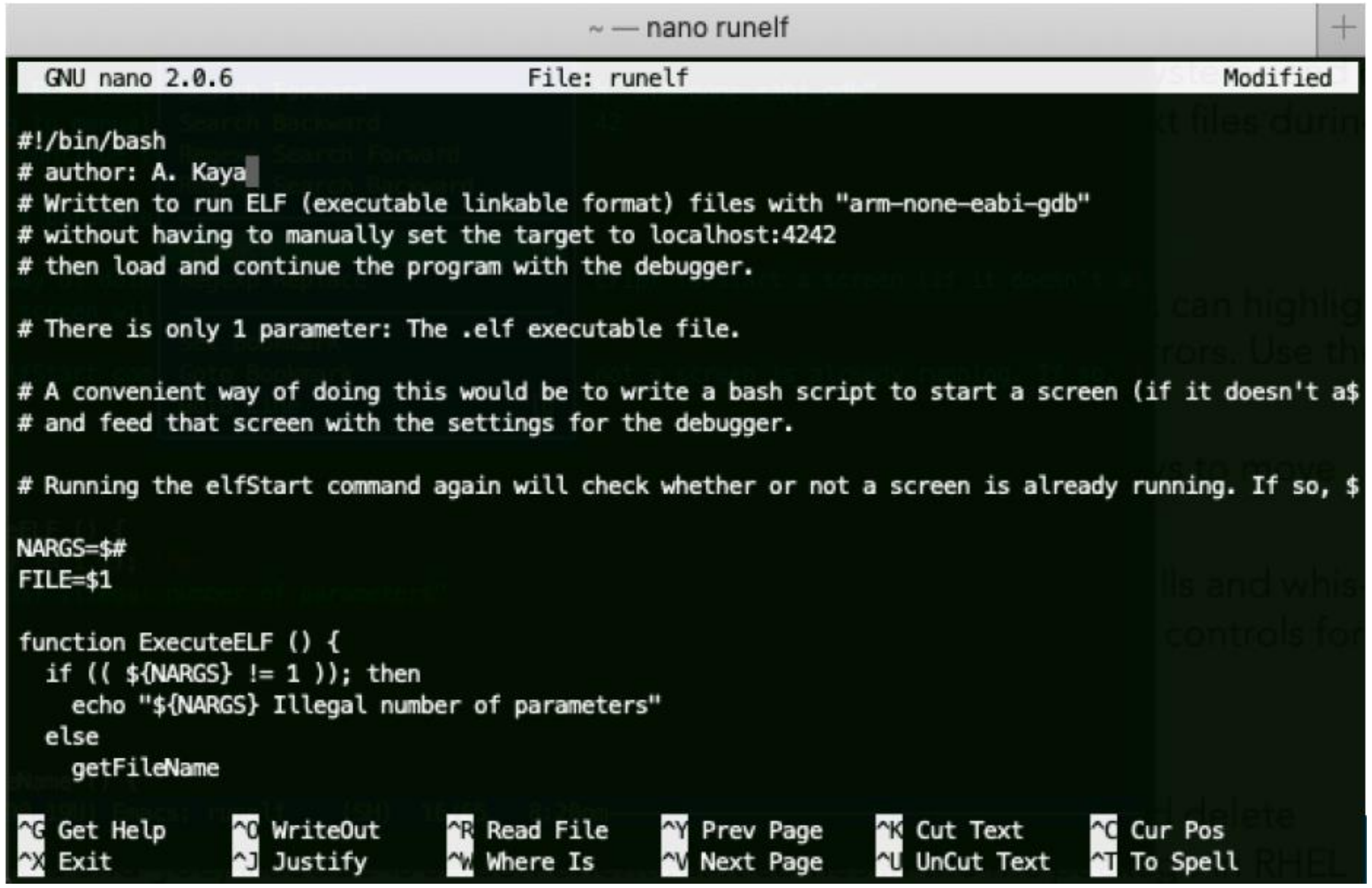
While some editors cover the basics, others are merely limited by your imagination...

User-friendly text editors:

- nano
- Jed

High input, high return editors:

- vim (vi improved)
- emacs (editor macros)



```
~ — nano runelf

 GNU nano 2.0.6                  File: runelf                          Modified

#!/bin/bash
# author: A. Kaya
# Written to run ELF (executable linkable format) files with "arm-none-eabi-gdb"
# without having to manually set the target to localhost:4242
# then load and continue the program with the debugger.

# There is only 1 parameter: The .elf executable file.

# A convenient way of doing this would be to write a bash script to start a screen (if it doesn't a$
# and feed that screen with the settings for the debugger.

# Running the elfStart command again will check whether or not a screen is already running. If so, $

NARGS=$#
FILE=$1

function ExecuteELF () {
  if (( ${NARGS} != 1 )); then
    echo "${NARGS} Illegal number of parameters"
  else
    getFileName

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Universiteit Antwerpen
Faculteit Toegepaste Ingenieurswetenschappen

Know your editors

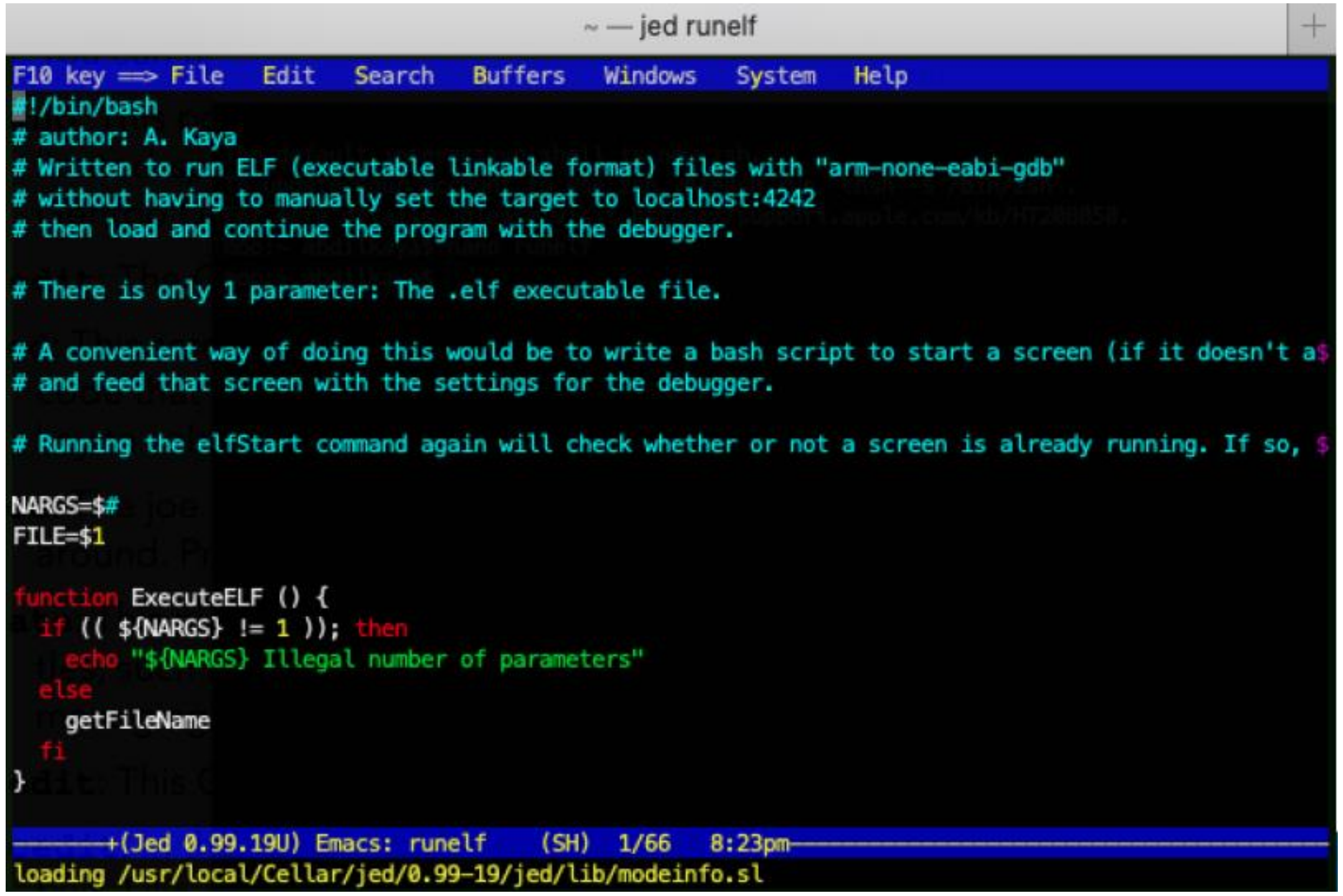
While some editors cover the basics, others are merely limited by your imagination...

User-friendly text editors:

- nano
- Jed

High input, high return editors:

- vim (vi improved)
- emacs (editor macros)



```
~ — jed runelf
F10 key ==> File   Edit   Search   Buffers   Windows   System   Help
#!/bin/bash
# author: A. Kaya
# Written to run ELF (executable linkable format) files with "arm-none-eabi-gdb"
# without having to manually set the target to localhost:4242
# then load and continue the program with the debugger.

# There is only 1 parameter: The .elf executable file.

# A convenient way of doing this would be to write a bash script to start a screen (if it doesn't a$
# and feed that screen with the settings for the debugger.

# Running the elfStart command again will check whether or not a screen is already running. If so, $

NARGS=$#
FILE=$1

function ExecuteELF () {
  if (( ${NARGS} != 1 )); then
    echo "${NARGS} Illegal number of parameters"
  else
    getFileName
  fi
}

-------+(Jed 0.99.19U) Emacs: runelf    (SH)  1/66   8:23pm------------------------------------------
loading /usr/local/Cellar/jed/0.99-19/jed/lib/modeinfo.sl
```

Universiteit Antwerpen | Faculteit Toegepaste Ingenieurswetenschappen

Know your editors

Try the vim tutorial, learn by using it

```
[student@localhost ~]$ sudo dnf install vim-enhanced
[student@localhost ~]$ vimtutor
```

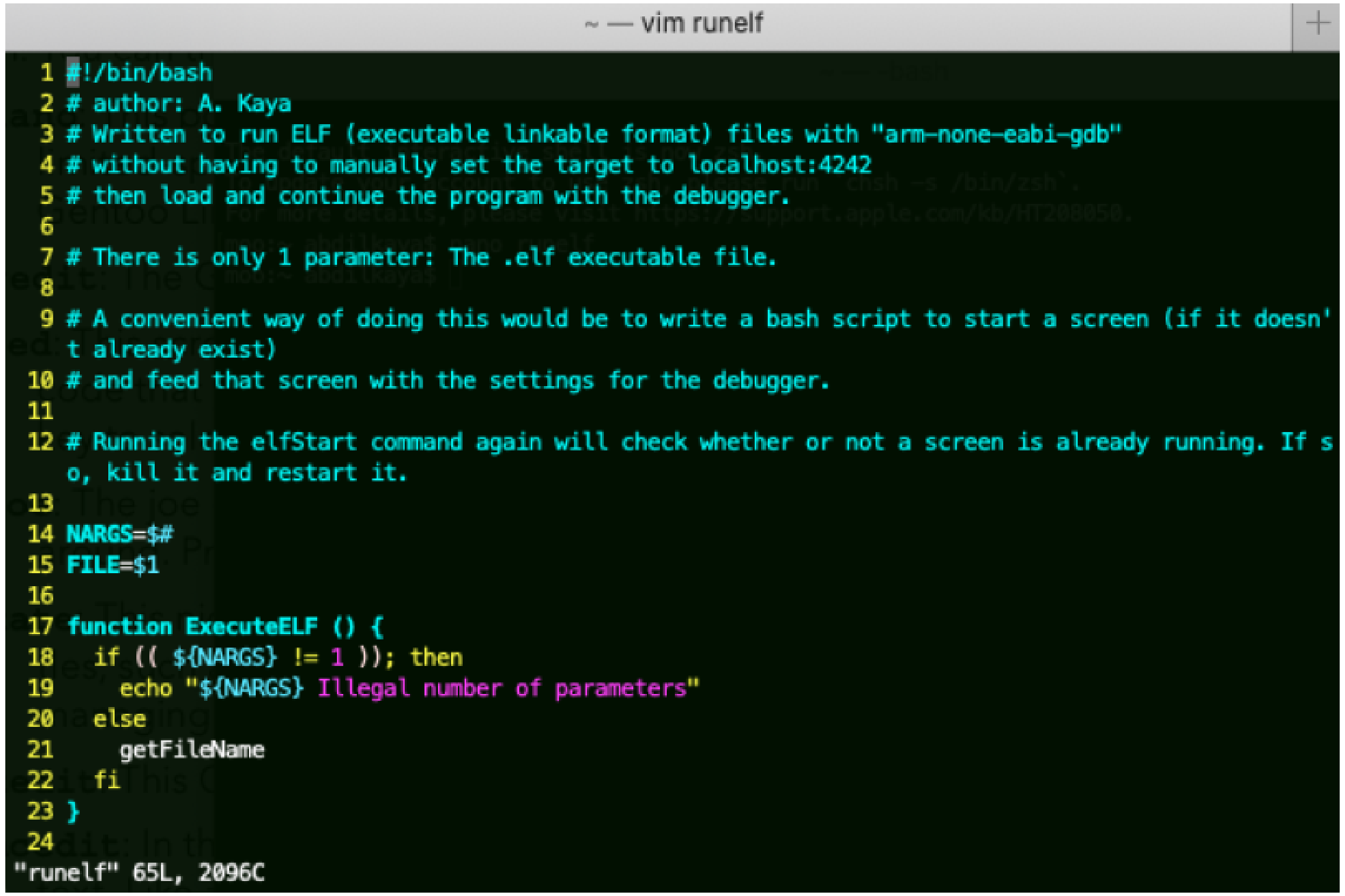
While some editors cover the basics, others are merely limited by your imagination...

User-friendly text editors:

- nano
- Jed

High input, high return editors:

- vim (vi improved)
- emacs (editor macros)



~ — vim runelf

```
#!/bin/bash
# author: A. Kaya
# Written to run ELF (executable linkable format) files with "arm-none-eabi-gdb"
# without having to manually set the target to localhost:4242
# then load and continue the program with the debugger.

# There is only 1 parameter: The .elf executable file.

# A convenient way of doing this would be to write a bash script to start a screen (if it doesn't already exist)
# and feed that screen with the settings for the debugger.

# Running the elfStart command again will check whether or not a screen is already running. If so, kill it and restart it.

NARGS=$#
FILE=$1

function ExecuteELF () {
  if (( ${NARGS} != 1 )); then
    echo "${NARGS} Illegal number of parameters"
  else
    getFileName
  fi
}

"runelf" 65L, 2096C
```

Know your editors

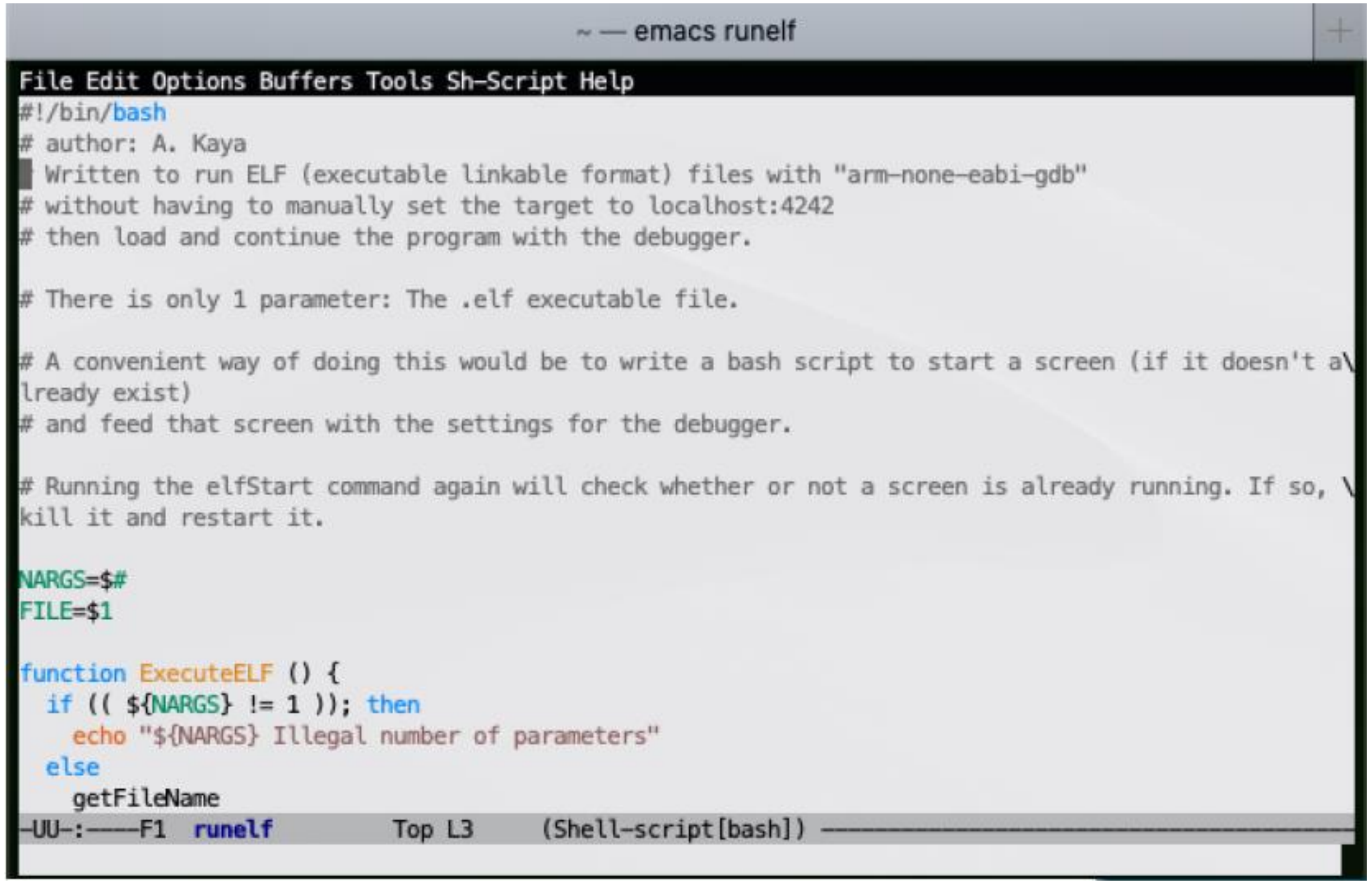
While some editors cover the basics, others are merely limited by your imagination...

User-friendly text editors:

- nano
- Jed

High input, high return editors:

- vim (vi improved)
- emacs (editor macros)



```
#!/bin/bash
# author: A. Kaya
# Written to run ELF (executable linkable format) files with "arm-none-eabi-gdb"
# without having to manually set the target to localhost:4242
# then load and continue the program with the debugger.

# There is only 1 parameter: The .elf executable file.

# A convenient way of doing this would be to write a bash script to start a screen (if it doesn't a\
lready exist)
# and feed that screen with the settings for the debugger.

# Running the elfStart command again will check whether or not a screen is already running. If so, \
kill it and restart it.

NARGS=$#
FILE=$1

function ExecuteELF () {
  if (( ${NARGS} != 1 )); then
    echo "${NARGS} Illegal number of parameters"
  else
    getFileName
-UU-:----F1  runelf          Top L3     (Shell-script[bash]) ------------------
```

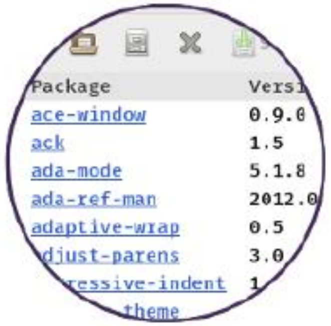
Should you learn vim or Emacs?

The keybindings of vim are such that you will --over time-- lose the need for a mouse or function keys. It is lightweight and often preinstalled.

Emacs on the other hand is more than just a text editor. It is an all-in-one workflow tool. And more...

```
▷ Org Agenda Dim Blocked Ta
  Non-nil means dim blocke

  Org Agenda Tags Todo Ho
  State : STANDARD.
  nil means hon
```

Highly customizable, using Emacs Lisp code or a graphical interface.



```
Package                Versi
ace-window             0.9.0
ack                    1.5
ada-mode               5.1.8
ada-ref-man            2012.0
adaptive-wrap          0.5
djust-parens           3.0
ressive-indent         1
  theme
```

A wide range of functionality beyond text editing, including a project planner, mail and news reader, debugger interface, calendar, IRC client, and more.

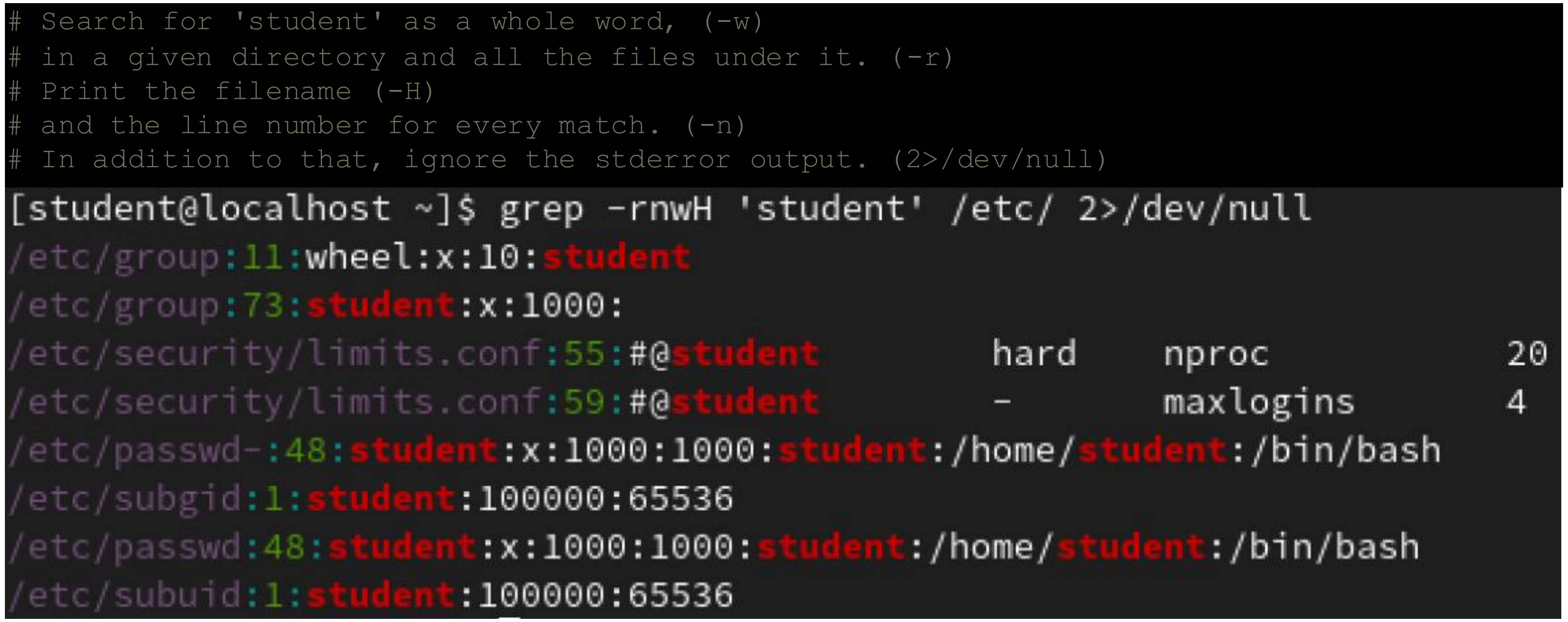
```
   ackages*
ee-mode is an avai

  Status: Available fro
 Archive: gnu

 Version: 0.4.1.1
 Summary: Major mode fo
Homepage: http://githu
eywords: coffeescri
```

A packaging system for downloading and installing extensions.

Commands to help you search

grep (globally search for a regular expression and print matching lines) : finds lines within files that match pattern

- A large amount of options that will help
- Useful for searching through piped outputs
- an example:



```
# Search for 'student' as a whole word, (-w)
# in a given directory and all the files under it. (-r)
# Print the filename (-H)
# and the line number for every match. (-n)
# In addition to that, ignore the stderror output. (2>/dev/null)
[student@localhost ~]$ grep -rnwH 'student' /etc/ 2>/dev/null
/etc/group:11:wheel:x:10:student
/etc/group:73:student:x:1000:
/etc/security/limits.conf:55:#@student        hard    nproc           20
/etc/security/limits.conf:59:#@student        -       maxlogins       4
/etc/passwd-:48:student:x:1000:1000:student:/home/student:/bin/bash
/etc/subgid:1:student:100000:65536
/etc/passwd:48:student:x:1000:1000:student:/home/student:/bin/bash
/etc/subuid:1:student:100000:65536
```

Universiteit Antwerpen
Faculteit Toegepaste Ingenieurswetenschappen

Commands to help you search

locate : finds by name in a database (daily update)

- Faster than **find**
- Cannot find files which have been added after the last database indexing
- Not all files are indexed. See configuration in /etc/updatedb.conf

find : the best find command to search by attributes

- You can immediately execute commands on the results using -exec
- Some attributes:
  - filename, ownership, permission, size, creation date, ...

TIP: Learn the basics of RegEx (Regular Expressions!

Managing running processes

Understanding processes

A process is an instance of a running program. A program can have many running instances.

This session will help you learn managing processes:

- launch, pause, stop or kill

using tools such as

- ps, top (or htop), kill, job

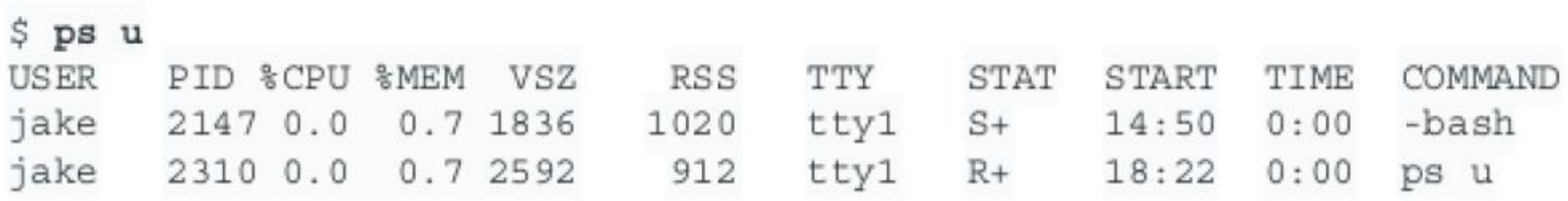
Processes are identified by:

- a unique process ID (PID), an associated user and group

Listing processes

ps : *"displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use top instead."*

- **ps u** : show the processes in this terminal for this user. *-u* means (show usernames)

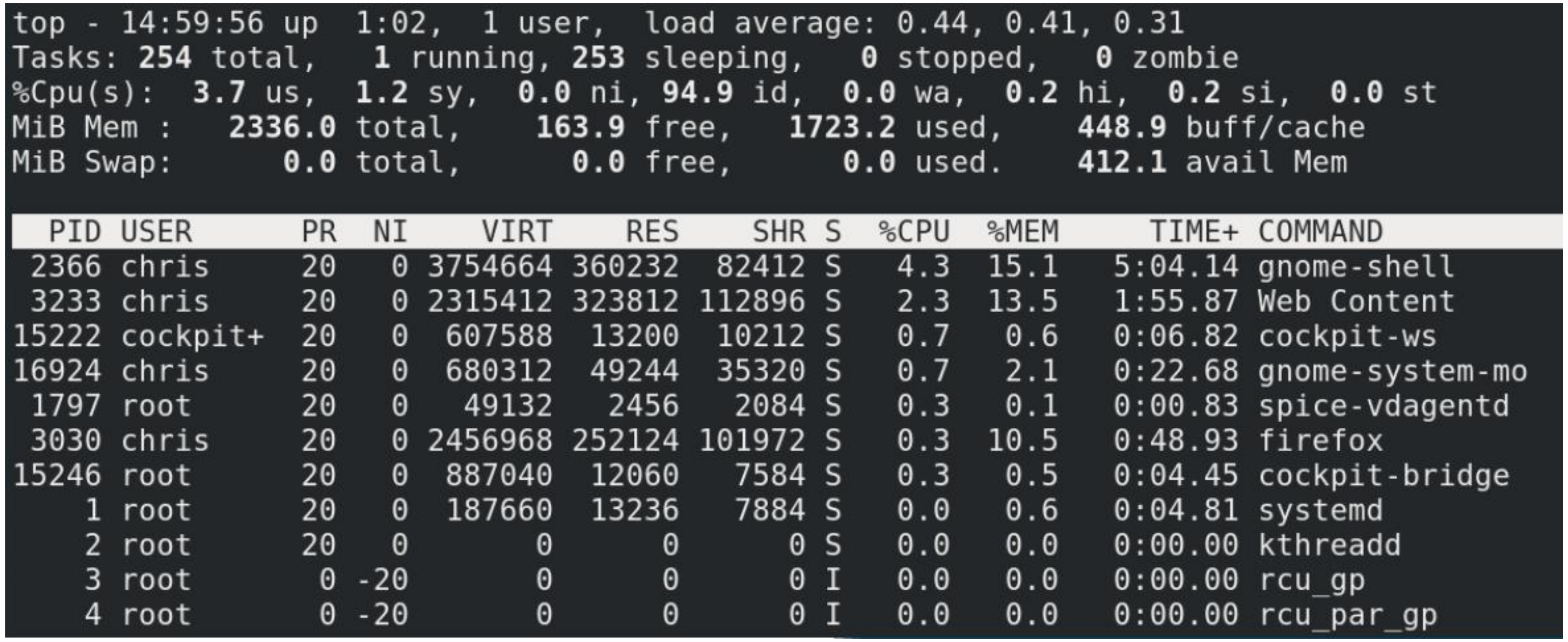


```
$ ps u
USER    PID %CPU %MEM  VSZ   RSS   TTY   STAT  START  TIME  COMMAND
jake    2147 0.0   0.7 1836  1020  tty1  S+    14:50  0:00  -bash
jake    2310 0.0   0.7 2592   912  tty1  R+    18:22  0:00  ps u
```

- pipe outputs (**|**) to **less** to scroll through the output e.g. **ps aux**
  - a: all users, u: print users, x: all terminals

Listing processes

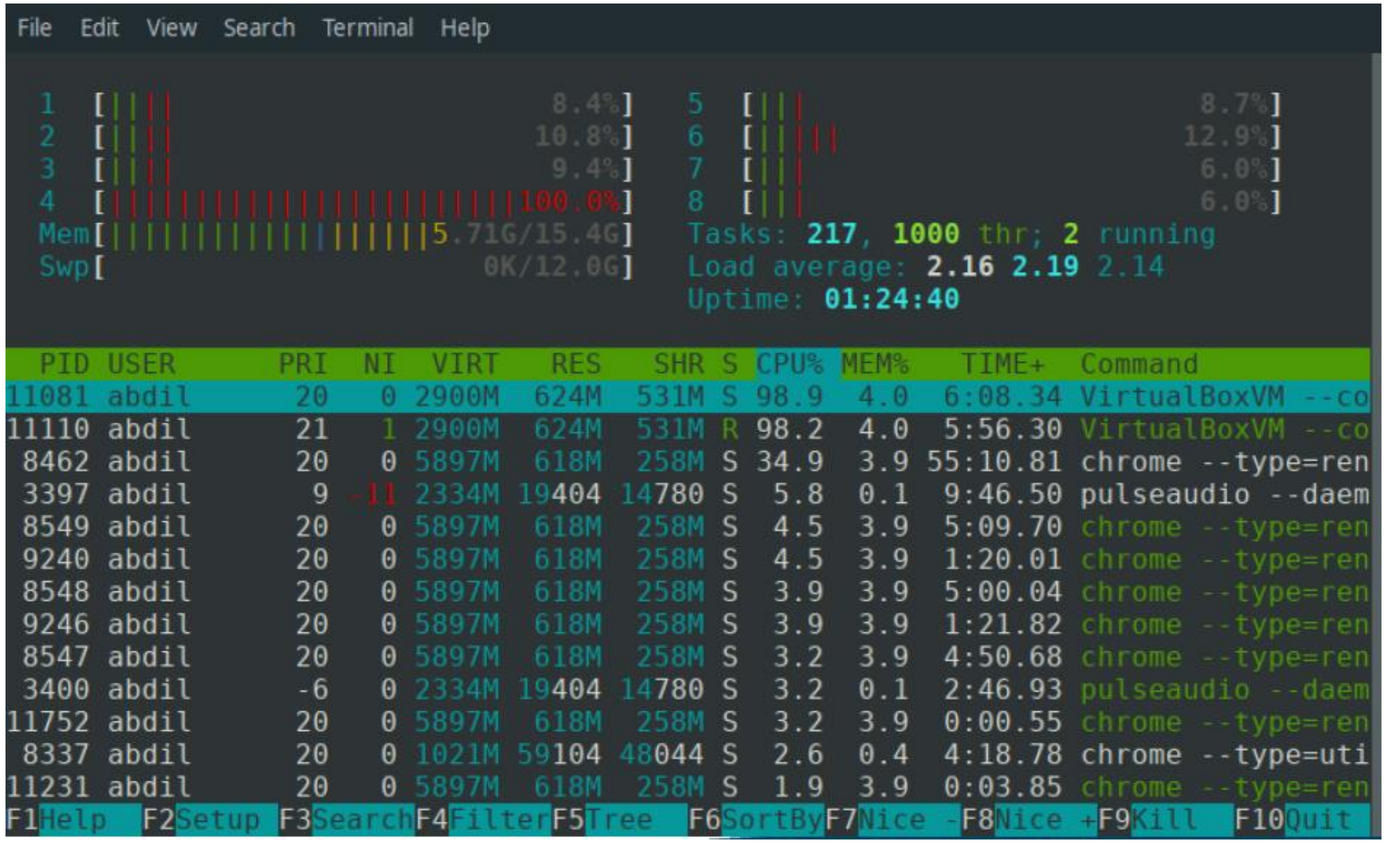
top : *"The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel. "*



```
top - 14:59:56 up  1:02,  1 user,  load average: 0.44, 0.41, 0.31
Tasks: 254 total,   1 running, 253 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.7 us,  1.2 sy,  0.0 ni, 94.9 id,  0.0 wa,  0.2 hi,  0.2 si,  0.0 st
MiB Mem :   2336.0 total,    163.9 free,   1723.2 used,    448.9 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    412.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 2366 chris     20   0 3754664 360232  82412 S   4.3  15.1   5:04.14 gnome-shell
 3233 chris     20   0 2315412 323812 112896 S   2.3  13.5   1:55.87 Web Content
15222 cockpit+  20   0  607588  13200  10212 S   0.7   0.6   0:06.82 cockpit-ws
16924 chris     20   0  680312  49244  35320 S   0.7   2.1   0:22.68 gnome-system-mo
 1797 root      20   0   49132   2456   2084 S   0.3   0.1   0:00.83 spice-vdagentd
 3030 chris     20   0 2456968 252124 101972 S   0.3  10.5   0:48.93 firefox
15246 root      20   0  887040  12060   7584 S   0.3   0.5   0:04.45 cockpit-bridge
    1 root      20   0  187660  13236   7884 S   0.0   0.6   0:04.81 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
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
```

Listing processes

htop : *"It is similar to top, but allows you to scroll vertically and horizontally"*

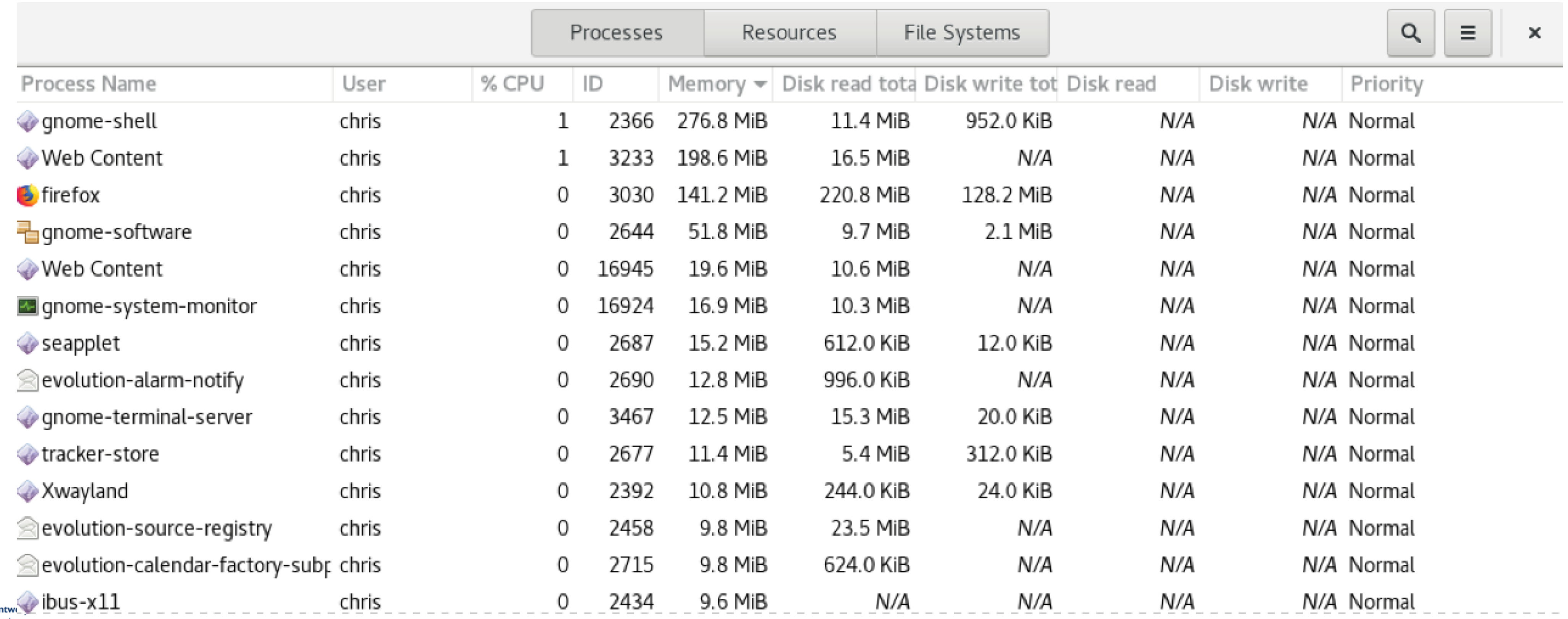


```
  1  [||||                      8.4%]   5  [|||                        8.7%]
  2  [||||                     10.8%]   6  [|||||                     12.9%]
  3  [||||                      9.4%]   7  [|||                        6.0%]
  4  [||||||||||||||||||||||||100.0%]   8  [|||                        6.0%]
  Mem[|||||||||||||||||||5.71G/15.4G]   Tasks: 217, 1000 thr; 2 running
  Swp[                     0K/12.0G]   Load average: 2.16 2.19 2.14
                                        Uptime: 01:24:40

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
11081 abdil      20   0 2900M  624M  531M S 98.9  4.0  6:08.34 VirtualBoxVM --co
11110 abdil      21   1 2900M  624M  531M R 98.2  4.0  5:56.30 VirtualBoxVM --co
 8462 abdil      20   0 5897M  618M  258M S 34.9  3.9 55:10.81 chrome --type=ren
 3397 abdil       9 -11 2334M 19404 14780 S  5.8  0.1  9:46.50 pulseaudio --daem
 8549 abdil      20   0 5897M  618M  258M S  4.5  3.9  5:09.70 chrome --type=ren
 9240 abdil      20   0 5897M  618M  258M S  4.5  3.9  1:20.01 chrome --type=ren
 8548 abdil      20   0 5897M  618M  258M S  3.9  3.9  5:00.04 chrome --type=ren
 9246 abdil      20   0 5897M  618M  258M S  3.9  3.9  1:21.82 chrome --type=ren
 8547 abdil      20   0 5897M  618M  258M S  3.2  3.9  4:50.68 chrome --type=ren
 3400 abdil      -6   0 2334M 19404 14780 S  3.2  0.1  2:46.93 pulseaudio --daem
11752 abdil      20   0 5897M  618M  258M S  3.2  3.9  0:00.55 chrome --type=ren
 8337 abdil      20   0 1021M 59104 48044 S  2.6  0.4  4:18.78 chrome --type=uti
11231 abdil      20   0 5897M  618M  258M S  1.9  3.9  0:03.85 chrome --type=ren
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

]

Listing processes

System Monitor : *"The System Monitor application displays a list of system processes, and monitors system usage. System Monitor shows which processes are running and how the processes are related. "*



Process Name	User	% CPU	ID	Memory	Disk read tota	Disk write tot	Disk read	Disk write	Priority
gnome-shell	chris	1	2366	276.8 MiB	11.4 MiB	952.0 KiB	N/A	N/A	Normal
Web Content	chris	1	3233	198.6 MiB	16.5 MiB	N/A	N/A	N/A	Normal
firefox	chris	0	3030	141.2 MiB	220.8 MiB	128.2 MiB	N/A	N/A	Normal
gnome-software	chris	0	2644	51.8 MiB	9.7 MiB	2.1 MiB	N/A	N/A	Normal
Web Content	chris	0	16945	19.6 MiB	10.6 MiB	N/A	N/A	N/A	Normal
gnome-system-monitor	chris	0	16924	16.9 MiB	10.3 MiB	N/A	N/A	N/A	Normal
seapplet	chris	0	2687	15.2 MiB	612.0 KiB	12.0 KiB	N/A	N/A	Normal
evolution-alarm-notify	chris	0	2690	12.8 MiB	996.0 KiB	N/A	N/A	N/A	Normal
gnome-terminal-server	chris	0	3467	12.5 MiB	15.3 MiB	20.0 KiB	N/A	N/A	Normal
tracker-store	chris	0	2677	11.4 MiB	5.4 MiB	312.0 KiB	N/A	N/A	Normal
Xwayland	chris	0	2392	10.8 MiB	244.0 KiB	24.0 KiB	N/A	N/A	Normal
evolution-source-registry	chris	0	2458	9.8 MiB	23.5 MiB	N/A	N/A	N/A	Normal
evolution-calendar-factory-subp	chris	0	2715	9.8 MiB	624.0 KiB	N/A	N/A	N/A	Normal
ibus-x11	chris	0	2434	9.6 MiB	N/A	N/A	N/A	N/A	Normal

Background and foreground processes

Run commands in the background by adding an ampersand (&) at the end. This will give you a job number (in brackets) and PID:

```
rnietvelt@3nwa:~$ find . 2>/dev/null > /tmp/homefolderfiles &
[1] 12826
```

Find current background jobs

```
rnietvelt@3nwa:~$ jobs
[1]+ Running find . 2> /dev/null > /tmp/homefolderfiles &
```

Bring it to the foreground by its job number

```
rnietvelt@3nwa:~$ fg %1
find . 2> /dev/null > /tmp/homefolderfiles
```

Killing processes with kill and killall

While **by default** used to send the terminate process signal (SIGTERM), the **kill** and **killall** commands can send any valid signal to a process.

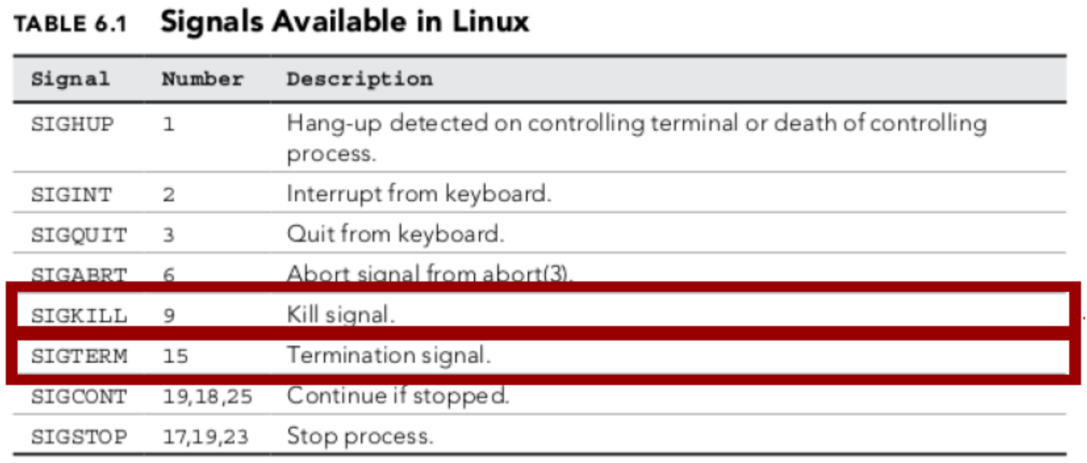


TABLE 6.1 Signals Available in Linux

Signal	Number	Description
SIGHUP	1	Hang-up detected on controlling terminal or death of controlling process.
SIGINT	2	Interrupt from keyboard.
SIGQUIT	3	Quit from keyboard.
SIGABRT	6	Abort signal from abort(3).
SIGKILL	9	Kill signal.
SIGTERM	15	Termination signal.
SIGCONT	19,18,25	Continue if stopped.
SIGSTOP	17,19,23	Stop process.

->more likely to actually kill a process than SIGTERM

The following three commands to terminate a process are identical:

```
kill 10905
kill -15 10905
kill -SIGTERM 10905
```

killall can be used to kill all instances of a program by its name.

```
killall -9 sometestprogram
```

Universiteit Antwerpen | Faculteit Toegepaste Ingenieurswetenschappen

"Niceness" of a process

The niceness of a process defines how much of the processor resources a program will claim. Claiming less makes a process nicer.

Range: **-20 to 19**

Default: **0**

- Regular users **can only increase** the niceness of their **own** processes.
- Regular users **cannot decrease** even if they initially increased.

Starting a process with a particular niceness level: use **nice**

```
nice -n +5 updatedb &
```

Changing the niceness of a running process: use **renice**

```
renice -n -5 20284
```

Exercises

Exercises

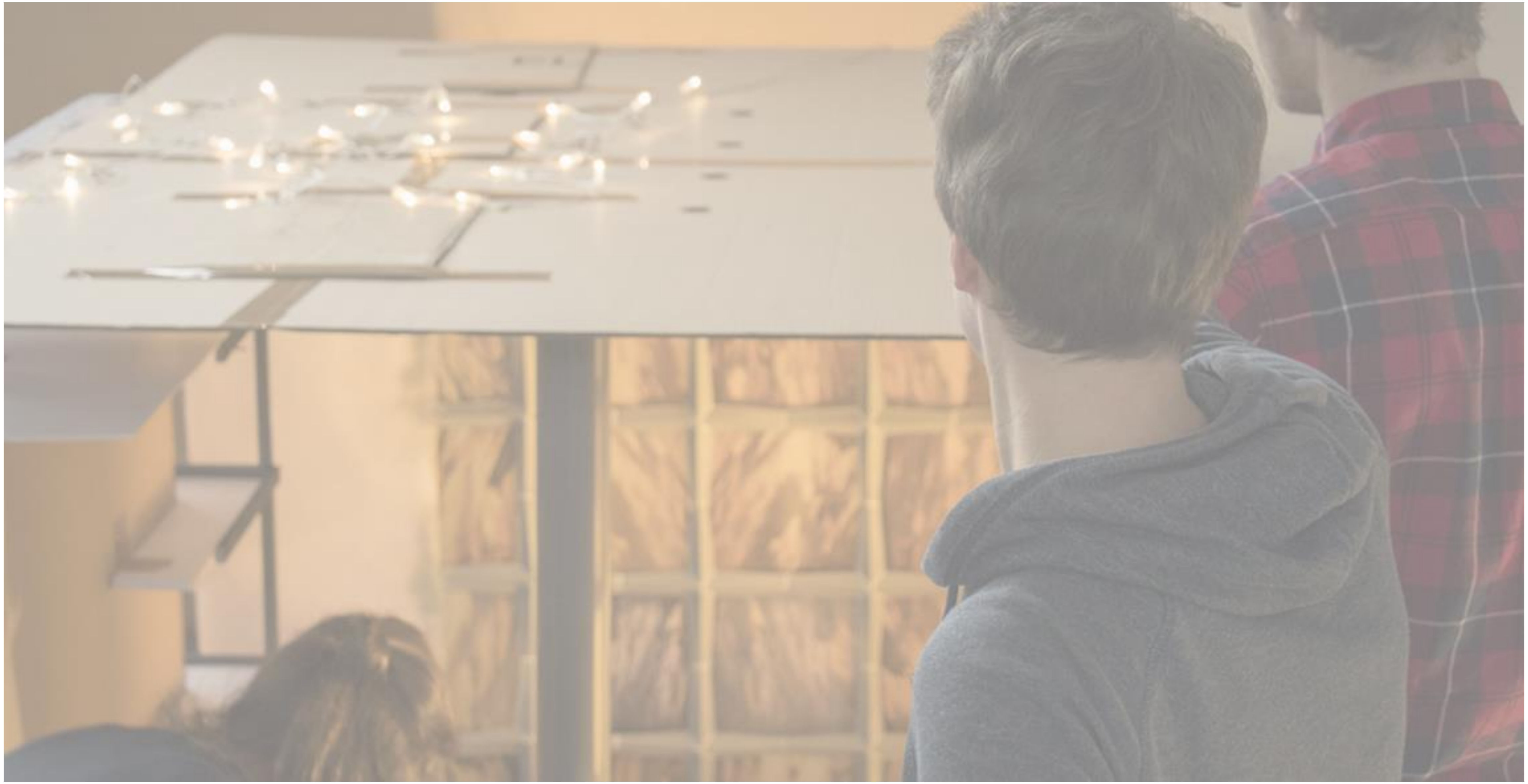
1. As a regular user, search the /usr/bin directory for every file named tty. Redirect error messages from your search to /dev/null.

2. Find every **file** in your user's home directory, and make a backup copy of each file in the same directory. Use each file's existing name, and just append **.bak** to create each backup file. This can be done in a single command line.

3. Find files under the /usr/bin directory that have not been modified in more than 10 years.

4. Run "yes > /dev/null" in the background. This time, using the kill command, send a signal to the process that causes it to pause (stop). Check the CPU usage (on for example the NetLab portal) before and after you pause the process. What does the command that you executed in the background do?

5. Use again the kill command to tell the process that you paused in the previous exercise, to continue working. Check the CPU usage again. Finally, permanently kill the process. Verify that it was killed.



Universiteit Antwerpen
Faculteit Toegepaste Ingenieurswetenschappen