Introduction to the Linux Operating System

Forschung und Wissenschaftliche Informationsversorgung
IT.SERVICES

RUHR
UNIVERSITÄT
BOCHUM

RUB

IT.SERVICES

RUB

Why Linux?

Why Linux?

RUB

GNU/Linux has several advantages over other systems:

- Reliable $\Rightarrow$ International space station is running Linux

Why Linux?

RUB

GNU/Linux has several advantages over other systems:

- Reliable $\Rightarrow$ International space station is running Linux
- Flexible $\Rightarrow$ Used from smart fridges to gaming consoles

Why Linux?

RUB

GNU/Linux has several advantages over other systems:

- Reliable ⇒ International space station is running Linux
- Flexible ⇒ Used from smart fridges to gaming consoles
- Scalable ⇒ Used on laptops and HPC-Clusters

Why Linux?

RUB

GNU/Linux has several advantages over other systems:

- Reliable $\Rightarrow$ International space station is running Linux
- Flexible $\Rightarrow$ Used from smart fridges to gaming consoles
- Scalable $\Rightarrow$ Used on laptops and HPC-Clusters
- GPL-licenced $\Rightarrow$ Open source and free to use

Why Linux?

RUB

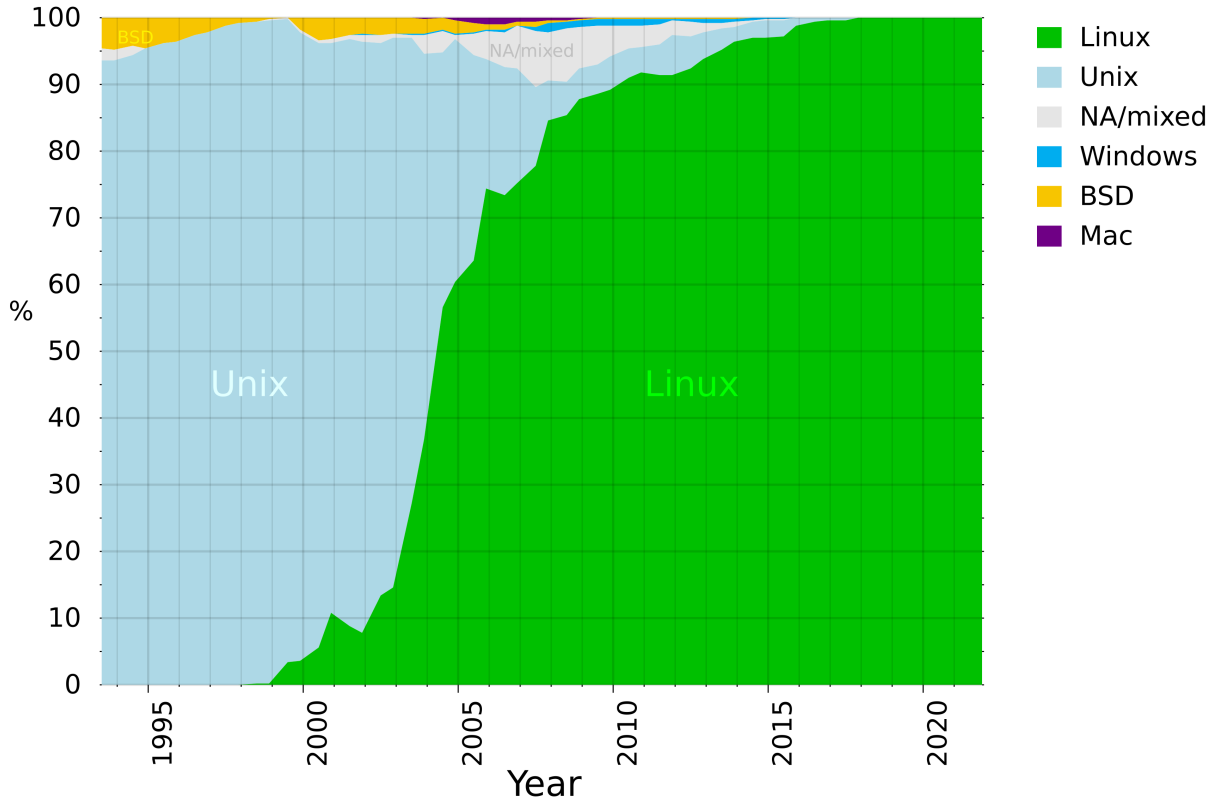
GNU/Linux has several advantages over other systems:

- Reliable ⇒ International space station is running Linux
- Flexible ⇒ Used from smart fridges to gaming consoles
- Scalable ⇒ Used on laptops and HPC-Clusters
- GPL-licenced ⇒ Open source and free to use
- Easy to use ⇒ Stay tuned

Linux: the HPC Operating System of Choice

RUB

TOP-500[1] fastest super computers operating systems over the years:



[1] https://commons.wikimedia.org/wiki/File:Operating_systems_used_on_top_500_supercomputers.svg

Why Linux?

RUB

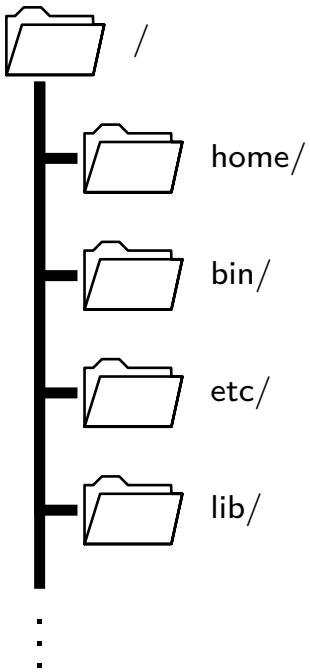
There is no way to use modern high performance computing resources without basic Linux knowledge!

Directory Structure

Linux Directory Structure

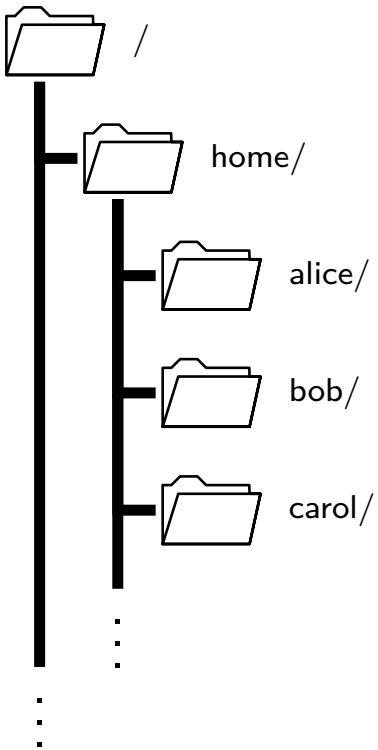
- Linux organizes directories and files in a hierarchical tree structure.

Linux Directory Structure



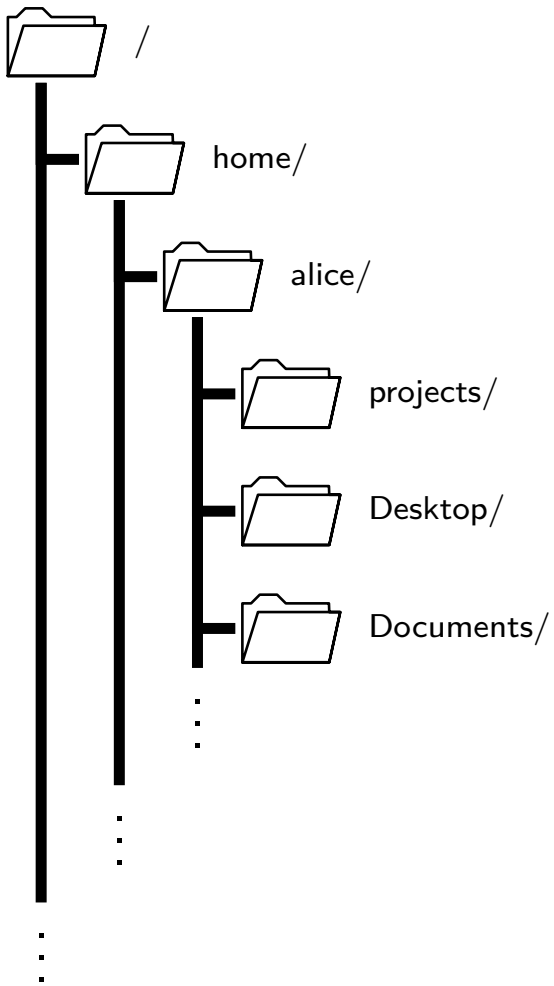
- Linux organizes directories and files in a hierarchical tree structure.
- Directories can contain more directories.

Linux Directory Structure



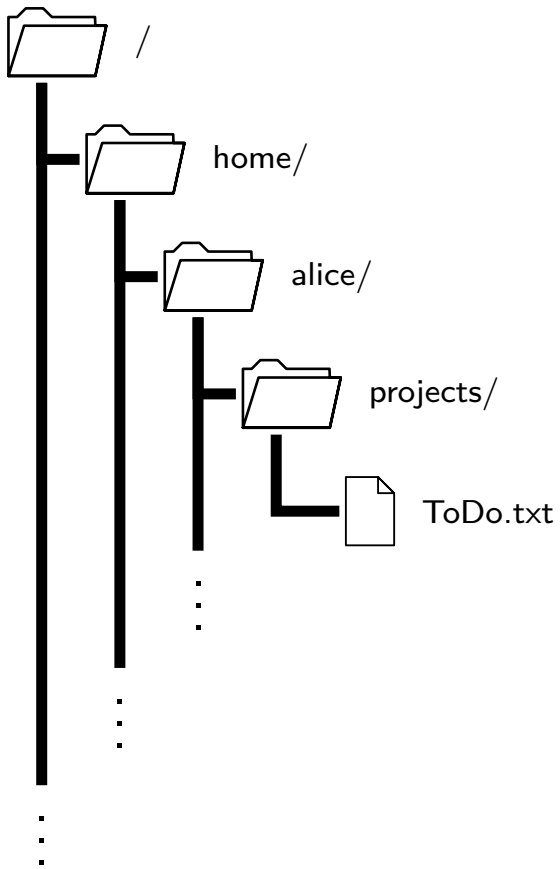
- Linux organizes directories and files in a hierarchical tree structure.
- Directories can contain more directories.
- The home directory contains one directory for every user.

Linux Directory Structure



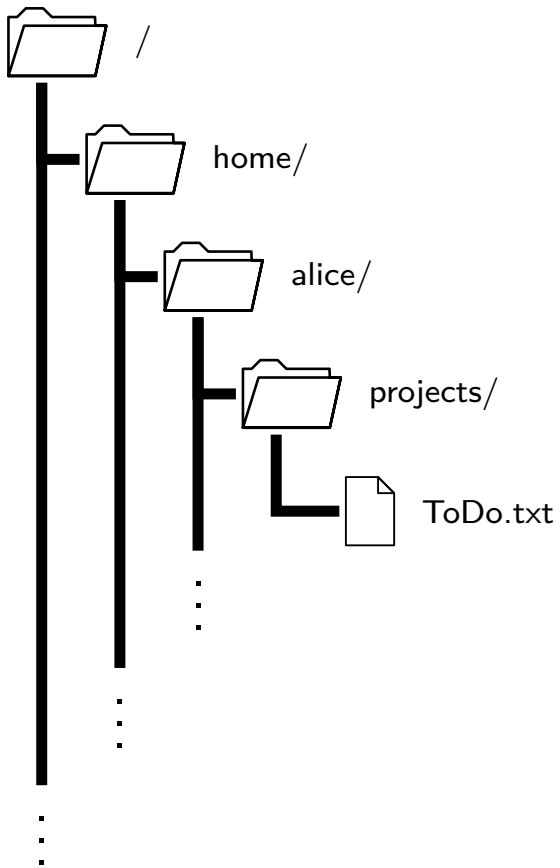
- Linux organizes directories and files in a hierarchical tree structure.
- Directories can contain more directories.
- The home directory contains one directory for every user.
- It contains the user owned directories and files.

Linux Directory Structure



- Linux organizes directories and files in a hierarchical tree structure.
- Directories can contain more directories.
- The home directory contains one directory for every user.
- It contains the user owned directories and files.
- The location of a file/directory can be written as a path through the tree:
  `/home/alice/projects/ToDo.txt`
  short: `~/projects/ToDo.txt`

Linux Directory Structure



- Linux organizes directories and files in a hierarchical tree structure.
- Directories can contain more directories.
- The home directory contains one directory for every user.
- It contains the user owned directories and files.
- The location of a file/directory can be written as a path through the tree:
  `/home/alice/projects/ToDo.txt`
  short: `~/projects/ToDo.txt`
- Note: Linux→/, Windows→\

The Terminal

The Terminal

RUB

Terminal

```
alice@hpc:$
```

- A terminal is a command line tool

The Terminal

RUB

Terminal

```
alice@hpc:$ hostname
```

- A terminal is a command line tool
- Users type a command

The Terminal

RUB

Terminal

```
alice@hpc:$ hostname
hpc
alice@hpc:$
```

- A terminal is a command line tool
- Users type a command
- Upon pressing 

the OS executes the command
- The result is printed to the terminal
- The terminal is ready for the next command

The Terminal

RUB

Terminal

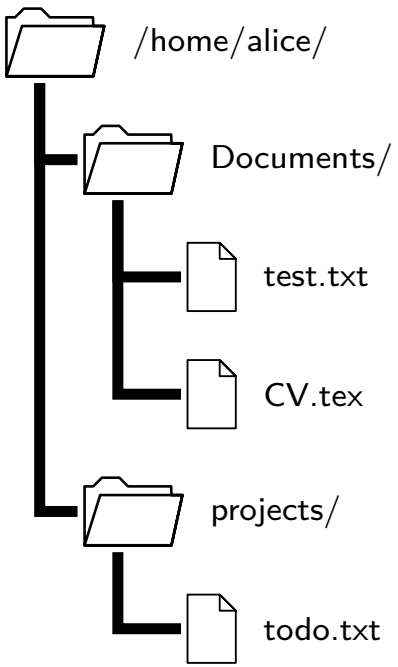
```
alice@hpc:$ hostname
hpc
alice@hpc:$
```

- A terminal is a command line tool
- Users type a command
- Upon pressing 

the OS executes the command
- The result is printed to the terminal
- The terminal is ready for the next command
- The  key can be used to autocomplete commands

Navigating the Directory Structure

Navigating the Directory Structure

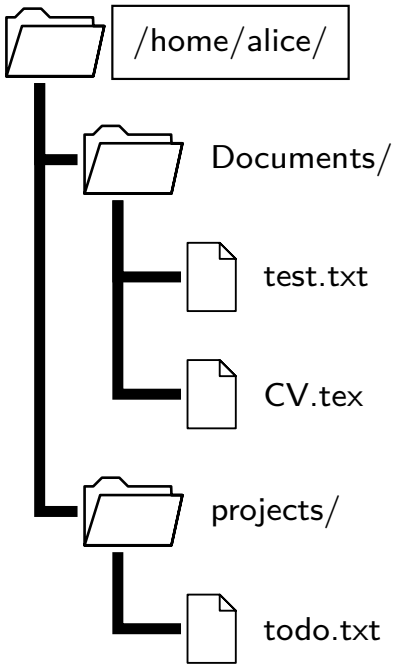


To navigate the directory tree in a terminal programs are executed.

Terminal

```
alice@hpc:$
```

Navigating the Directory Structure

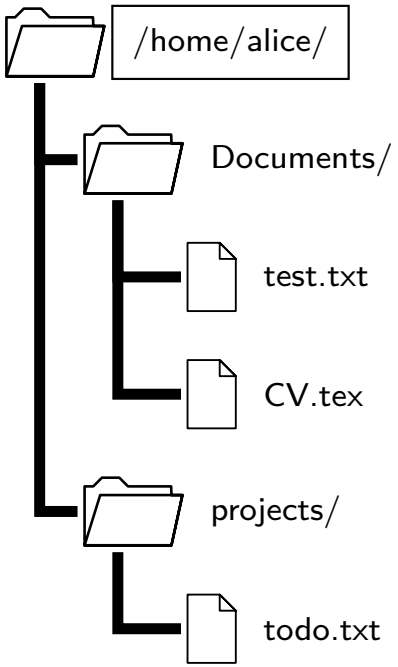


Which folder am I currently in?
`pwd` (print working directory).

Terminal

```
alice@hpc:$ pwd
/home/alice
```

Navigating the Directory Structure

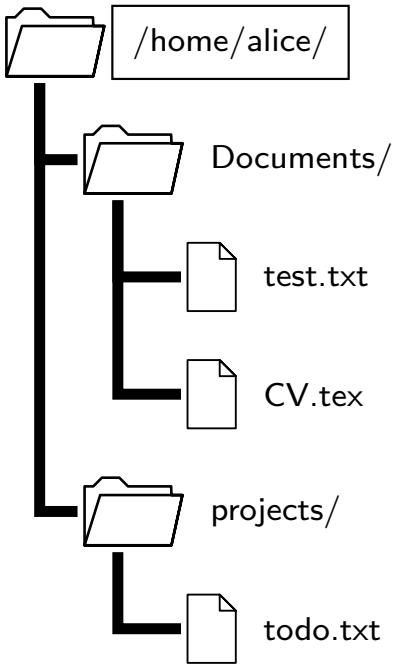


What is contained in the folder?
`ls` (list contents of current directory).

Terminal

```
alice@hpc:$ ls
Documents projects
```

Navigating the Directory Structure



Some programs take arguments.

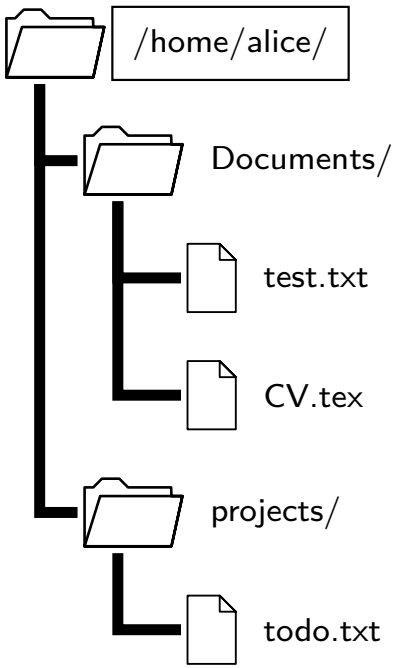
`ls Documents`

(list contents of "`Documents`").

Terminal

```
alice@hpc:$ ls Documents
CV.tex test.txt
```

Navigating the Directory Structure

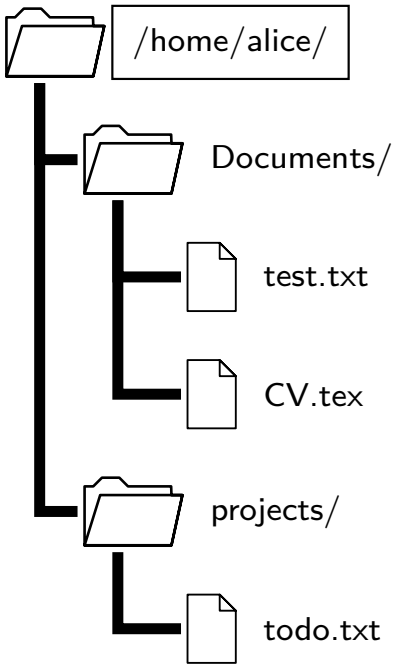


Tune program behavior with flags.
`ls -l` (list long-format) lists files and directories with extra information.

Terminal

```
alice@hpc:$ ls -l
total 8
drwxrwxr-x 2 alice alice 4096 Feb 28 10:49 Documents
drwxrwxr-x 2 alice alice 4096 Feb 28 11:12 projects
```

Navigating the Directory Structure

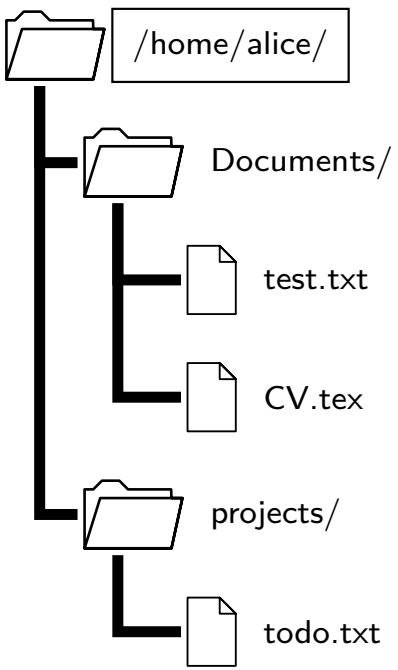


`ls -lt` or `ls -l -t` (list long-format time-sorted) sorts the list by time stamp.

Terminal

```
alice@hpc:$ ls -lt
total 8
drwxrwxr-x 2 alice alice 4096 Feb 28 11:12 projects
drwxrwxr-x 2 alice alice 4096 Feb 28 10:49 Documents
```

Navigating the Directory Structure



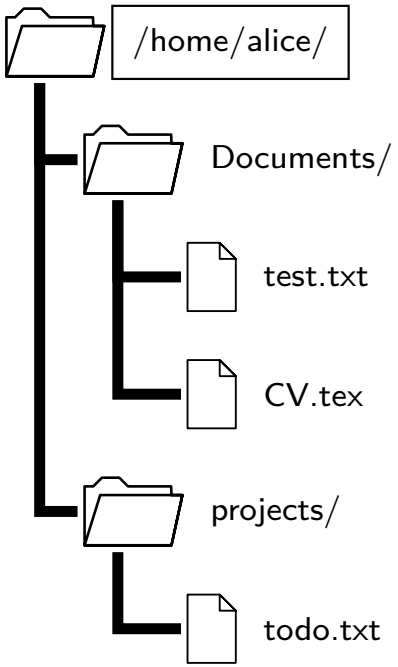
Many programs have a special `--help` flag to show how to use it.

`ls --help` (shows help information for the ls program)

Terminal

```
alice@hpc:$ ls --help
Usage:  ls [OPTION]...  [FILE]...
List information about the FILEs (the current
directory by default).
...
```

Navigating the Directory Structure

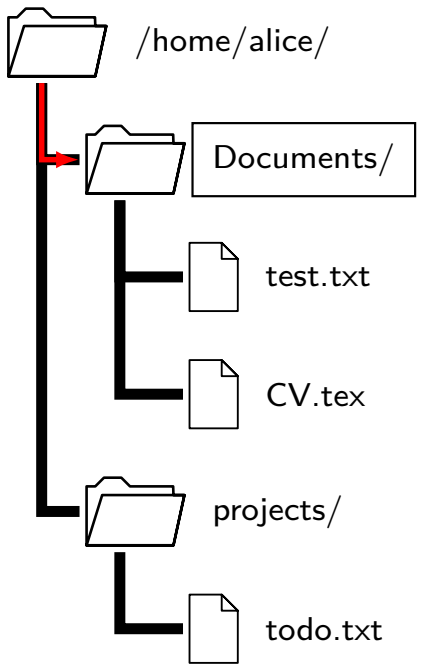


How to change the current directory?

Terminal

```
alice@hpc:$ pwd
/home/alice/
```

Navigating the Directory Structure

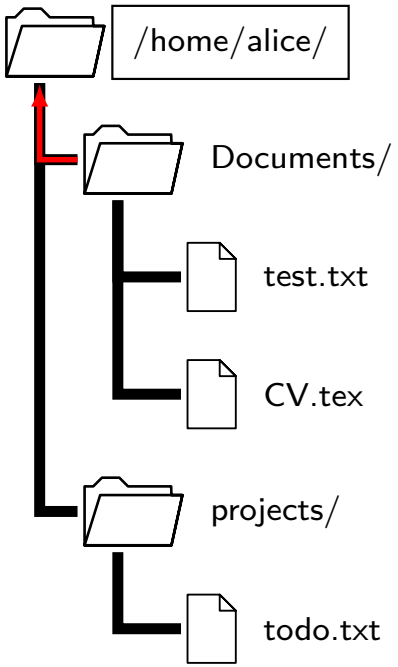


`cd Documents` (change directory) will change the current directory to "Documents" (if it exists).

Terminal

```
alice@hpc:$ cd Documents
alice@hpc:$ pwd
/home/alice/Documents
```

Navigating the Directory Structure

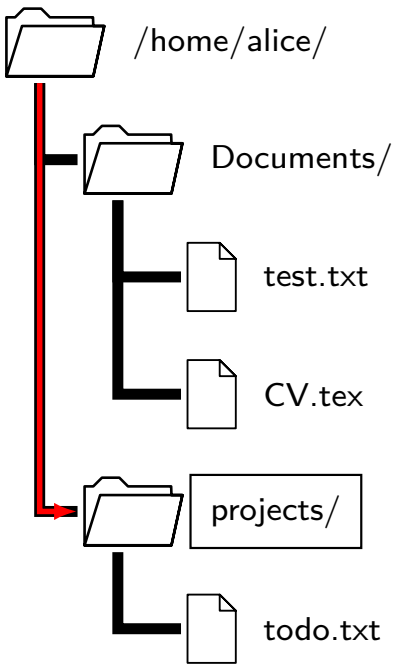


`cd ..` will leave the folder and go to the next higher level.

Terminal

```
alice@hpc:$ cd ..
alice@hpc:$ pwd
/home/alice
```

Navigating the Directory Structure

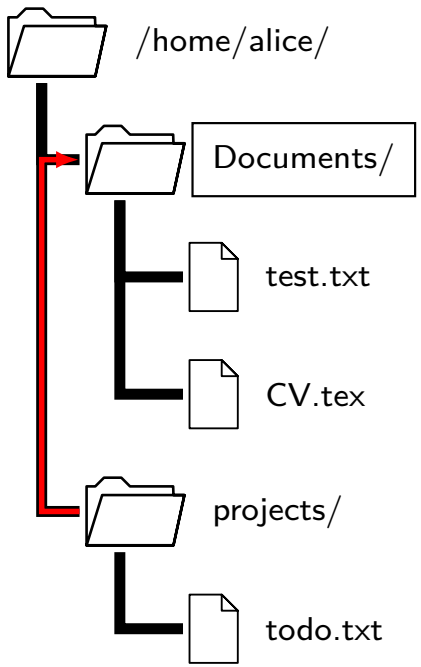


`cd /home/alice/projects/` will go to that folder (absolute paths start with /).

Terminal

```
alice@hpc:$ cd /home/alice/projects/
alice@hpc:$ pwd
/home/alice/projects
```

Navigating the Directory Structure

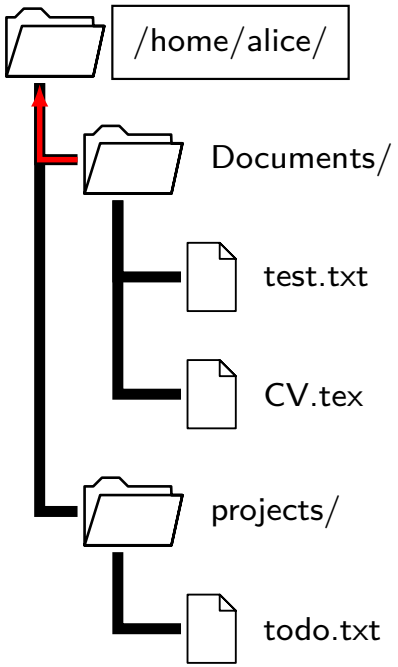


`cd ../Documents/` will go one level up then into “Documents” (relative paths do not start with /)

Terminal

```
alice@hpc:$ cd ../Documents
alice@hpc:$ pwd
/home/alice/Documents
```

Navigating the Directory Structure



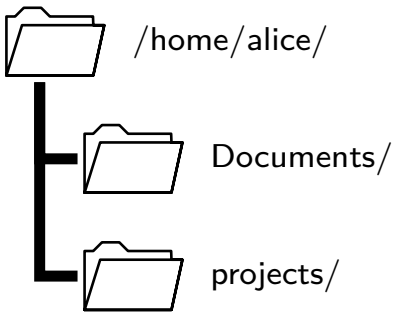
`cd ~` will go to /home/alice or whatever your home directory is.

Terminal

```
alice@hpc:$ cd ~
alice@hpc:$ pwd
/home/alice
```

Modifying the Directory Structure

Modifying the Directory Structure



`mkdir inputs` (make directory) creates a directory with name “input”.

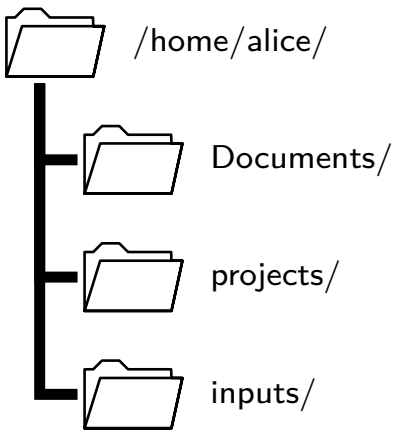
Terminal

```
alice@hpc:$ ls
Documents projects
```

Warning!

Linux deletes files/directories without asking for confirmation. It assumes you know what you are doing.

Modifying the Directory Structure



`mkdir inputs` (make directory) creates a directory with name “input”.

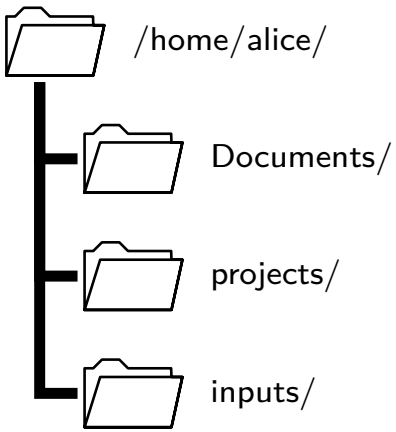
Terminal

```
alice@hpc:$ mkdir inputs
```

Warning!

Linux deletes files/directories without asking for confirmation. It assumes you know what you are doing.

Modifying the Directory Structure



`mkdir inputs` (make directory) creates a directory with name “input”.

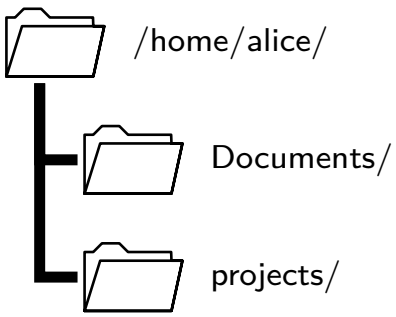
Terminal

```
alice@hpc:$ ls
Documents projects inputs
```

Warning!

Linux deletes files/directories without asking for confirmation. It assumes you know what you are doing.

Modifying the Directory Structure



`rmdir inputs` (remove directory) deletes the directory input if it is empty.

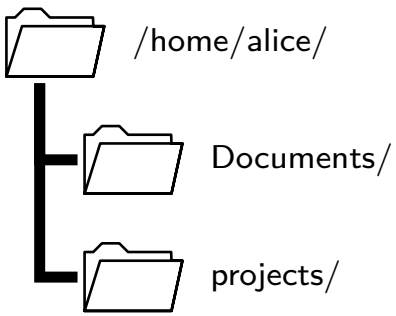
Terminal

```
alice@hpc:$ rmdir inputs
```

Warning!

Linux deletes files/directories without asking for confirmation. It assumes you know what you are doing.

Modifying the Directory Structure



`rmdir inputs` (remove directory) deletes the directory input if it is empty.

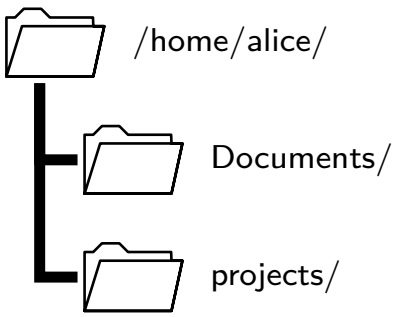
Terminal

```
alice@hpc:$ ls
Documents projects
```

Warning!

Linux deletes files/directories without asking for confirmation. It assumes you know what you are doing.

Modifying the Directory Structure



`rm -r inputs` (remove recursively) deletes a file/directory with name "inputs" and all its contents.

Terminal

```
alice@hpc:$ rm -r inputs
```

Warning!

Linux deletes files/directories without asking for confirmation. It assumes you know what you are doing.

Exercise 1

1. Open a Terminal
   - Ctrl + Alt + T
   - Via the startmenu
2. Check your current location
3. Create a new folder named `linuxintro`
4. Change into that folder and check your current location

Exercise 1

RUB

1. Open a Terminal
   - 
   - Via the startmenu

Terminal

```
alice@hpc:$
```

Exercise 1

1. Open a Terminal
   - 
   - Via the startmenu
2. Check your current location

Terminal

```
alice@hpc:$ pwd
/home/alice
alice@hpc:$
```

Exercise 1

RUB

1. Open a Terminal
   - Ctrl + Alt + T
   - Via the startmenu
2. Check your current location
3. Create a new folder named `linuxintro`

Terminal

```
alice@hpc:$ pwd
/home/alice
alice@hpc:$ mkdir linuxintro
alice@hpc:$
```

Exercise 1

RUB

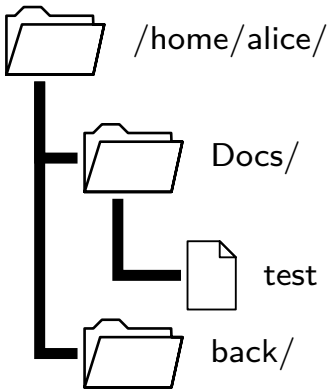
1. Open a Terminal
   - Ctrl + Alt + T
   - Via the startmenu
2. Check your current location
3. Create a new folder named `linuxintro`
4. Change into that folder and check your current location

Terminal

```
alice@hpc:$ pwd
/home/alice
alice@hpc:$ mkdir linuxintro
alice@hpc:$ cd linuxintro
alice@hpc:$ pwd
/home/alice/linuxintro
```

Handling Files

Handling Files



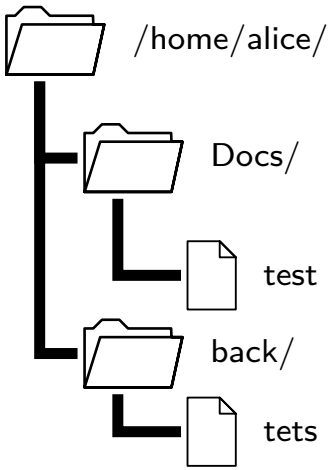
Terminal

```
alice@hpc:$ ls Docs
test
alice@hpc:$ ls back
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



`cp Docs/test back/tets`
(copy Docs/test to back/tets).

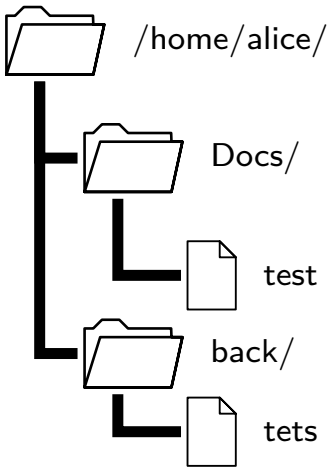
Terminal

```
alice@hpc:$ cp Docs/test back/tets
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



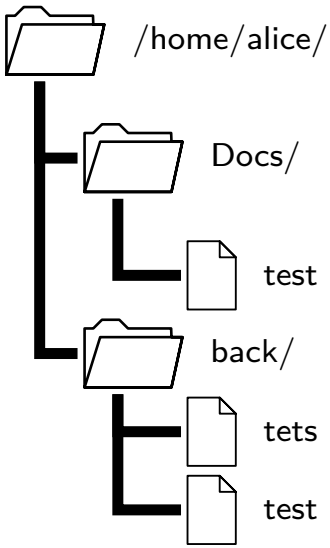
Terminal

```
alice@hpc:$ ls Docs
test
alice@hpc:$ ls back
tets
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



`cp Docs/test back/`
(copy Docs/test into directory back/).

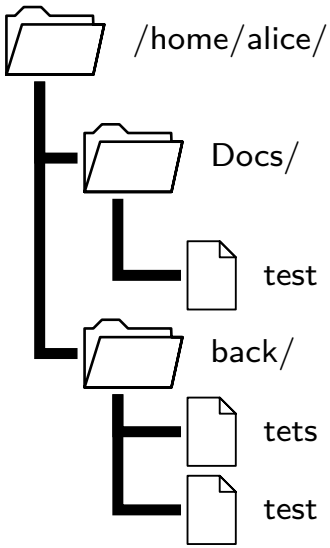
Terminal

```
alice@hpc:$ cp Docs/test back/
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



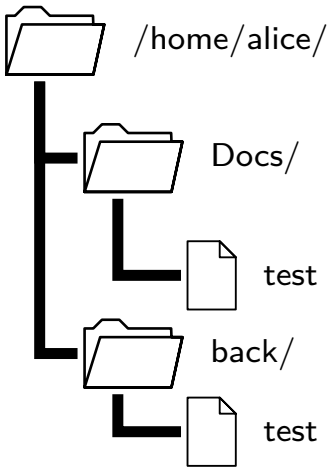
Terminal

```
alice@hpc:$ ls Docs
test
alice@hpc:$ ls back
test tets
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



`rm back/tets`
(remove back/tets).

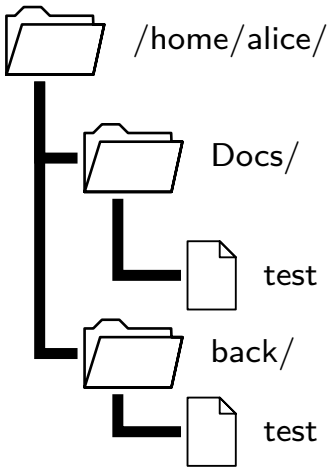
Terminal

```
alice@hpc:$ rm back/tets
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



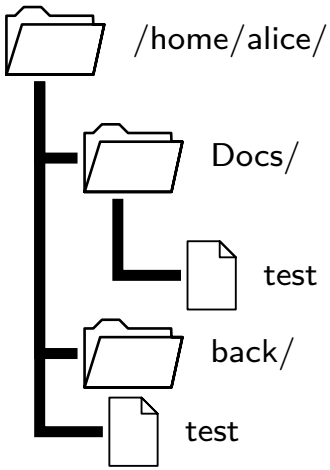
Terminal

```
alice@hpc:$ ls Docs
test
alice@hpc:$ ls back
test
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



`mv back/test ./test`
(move back/test to ./test).

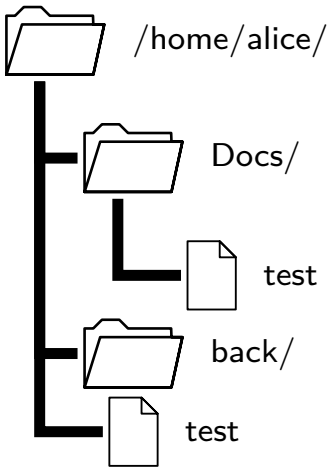
Terminal

```
alice@hpc:$ mv back/test ./test
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Handling Files



Terminal

```
alice@hpc:$ ls Docs
test
alice@hpc:$ ls back
alice@hpc:$ ls
test
```

Warning!

Linux overwrites/deletes files without asking for confirmation.
It assumes you know what you are doing.

Exercise 2

1. Use the `--help` flag to learn how to use `touch` to create a new empty file
2. Create two new empty files named `data.txt` and `script.sh`

Exercise 2

1. Use the `--help` flag to learn how to use `touch` to create a new empty file

Terminal

```
alice@hpc:$ touch --help
Usage:  touch [OPTION]...  FILE...
Update the access and modification times of each FILE to the current time.

A FILE argument that does not exist is created empty, unless -c or -h
is supplied.
```

Exercise 2

2. Create two new empty files named `data.txt` and `script.sh`

Terminal

```
alice@hpc:$ touch data.txt
alice@hpc:$ touch script.sh
alice@hpc:$ ls
data.txt script.sh
```

Permission Denied

Permission Denied

RUB

Terminal

```
alice@hpc:$ ls /root/
ls: cannot open directory '/root/': Permission denied
```

- Users do not have permission to access every file/directory

Permission Denied

RUB

Terminal

```
alice@hpc:$ ls -l /
lrwxrwxrwx   1 root root     7 Jan  3  2023 bin -> usr/bin
drwxr-xr-x   3 root root  4096 Jan  3  2023 home
drwx------   5 root root  4096 Dez 22 14:51 root
-rw-r--r--  18 root root    96 Jan 30 08:24 afile
```

- Users do not have permission to access every file/directory

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Owning **user** of the file/directory

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Owning **user** of the file/directory
- Owning **group** of the file/directory

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “-“→file, “**l**“→link)

- Owning **user** of the file/directory
- Owning **group** of the file/directory

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “**-**“→file, “**l**“→link)
- Permissions (“**r**“→read, “**w**“→write, “**x**“→execute)

- Owning **user** of the file/directory
- Owning **group** of the file/directory

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “**-**“→file, “**l**“→link)
- Permissions (“**r**“→read, “**w**“→write, “**x**“→execute)
  - Permissions for owning **user**
- Owning **user** of the file/directory
- Owning **group** of the file/directory
- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “-“→file, “**l**“→link)
- Permissions (“**r**“→read, “**w**“→write, “**x**“→execute)
  - Permissions for owning **user**
  - Permissions for owning **group**

- Owning **user** of the file/directory
- Owning **group** of the file/directory

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “**-**“→file, “**l**“→link)
- Permissions (“**r**“→read, “**w**“→write, “**x**“→execute)
  - Permissions for owning **user**
  - Permissions for owning **group**
  - Permissions for everyone else (**others**)

- Owning **user** of the file/directory
- Owning **group** of the file/directory

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “**-**“→file, “**l**“→link)
- Permissions (“**r**“→read, “**w**“→write, “**x**“→execute)
  - Permissions for owning **user**
  - Permissions for owning **group**
  - Permissions for everyone else (**others**)

- Owning **user** of the file/directory
- Owning **group** of the file/directory
- Size in bytes

- File/Directory name

Reading Permissions

```
-rwxr-xr-x   1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “**-**“→file, “**l**“→link)
- Permissions (“**r**“→read, “**w**“→write, “**x**“→execute)
  - Permissions for owning **user**
  - Permissions for owning **group**
  - Permissions for everyone else (**others**)

- Owning **user** of the file/directory
- Owning **group** of the file/directory
- Size in bytes
- Date of last modification
- File/Directory name

Reading Permissions

```
-rwxr-xr-x  1 alice phys  4096 Jan  3  2023 my_project
```

- Type (“**d**“→directory, “**-**“→file, “**l**“→link)
- Permissions (“**r**“→read, “**w**“→write, “**x**“→execute)
  - Permissions for owning **user**
  - Permissions for owning **group**
  - Permissions for everyone else (**others**)
- Number of hard links
- Owning **user** of the file/directory
- Owning **group** of the file/directory
- Size in bytes
- Date of last modification
- File/Directory name

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.
a: u→user, g→group, o→other

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.
a: u→user, g→group, o→other
±: +→add permission, −→remove permission

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod u+x my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod u+x my_project
alice@hpc:$ ls -l
-rwx-w-r--   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod g+r my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod g+r my_project
alice@hpc:$ ls -l
-rw-rw-r--   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod o+wx my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod o+wx my_project
alice@hpc:$ ls -l
-rw--w-rwx   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod u-w my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod u-w my_project
alice@hpc:$ ls -l
-r---w-r--   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod g-w my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod g-w my_project
alice@hpc:$ ls -l
-rw----r--   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod o-r my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod o-r my_project
alice@hpc:$ ls -l
-rw--w----   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod u+rwx,g+rwx,o-rwx my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod u+rwx,g+rwx,o-rwx my_project
alice@hpc:$ ls -l
-rwxrwx---   1 alice phys ... my_project
```

Setting Permissions

`chmod a±b filename` alters the permissions of a file/directory.

a: u→user, g→group, o→other

±: +→add permission, −→remove permission

b: r→read, w→write, x→execute

Terminal

```
alice@hpc:$ ls -l
-rw--w-r--   1 alice phys ... my_project
alice@hpc:$ chmod u+rwx,g+rwx,o-rwx my_project
alice@hpc:$ ls -l
-rwxrwx---   1 alice phys ... my_project
```

To change the owner use: `chown user:group filename`

Setting Permissions (Alternative)

Dec	Bin	Perm
0	000	`---`
1	001	`--x`
2	010	`-w-`
3	011	`-wx`
4	100	`r--`
5	101	`r-x`
6	110	`rw-`
7	111	`rwx`

Use `chmod UGO file` to set permissions.
(U→User, G→Group, O→Other)

Terminal

```
alice@hpc:$ chmod 643 my_project
alice@hpc:$ ls -l
-rw-r---wx-   1 alice phys ... my_project
```

User $6 = 4 + 2 \rightarrow$ `rw-`

Group $4 \rightarrow$ `r--`

Others $3 = 2 + 1 \rightarrow$ `-wx`

To change the owner use: `chown user:group filename`

Exercise 3

1. Change permissions of `script.sh` such that:
   - you can read, write, and execute
   - your group can read, and execute
   - others can only read

Exercise 3

1. Change permissions of `script.sh` such that:
   - you can read, write, and execute
   - your group can read, and execute
   - others can only read

Terminal

```
alice@hpc:$ chmod u+x,g+x,g-w script.sh
alice@hpc:$ ls -l
total 0
-rwxr-xr-- 1 alice alicegrp 0 Jun 17 08:12 script.sh
```

Editing Files

Editing Files

- Scripts, configurations, inputs, outputs, logs, ... are usually textfiles.

Editing Files

- Scripts, configurations, inputs, outputs, logs, ... are usually textfiles.
- Linux provides terminal based editors for textfiles in a huge variaty.

Editing Files

RUB

- Scripts, configurations, inputs, outputs, logs, ... are usually textfiles.
- Linux provides terminal based editors for textfiles in a huge variaty.
- Famous are:
  - emacs (powerful, complex)

Editing Files

RUB

- Scripts, configurations, inputs, outputs, logs, ... are usually textfiles.
- Linux provides terminal based editors for textfiles in a huge variaty.
- Famous are:
  - emacs (powerful, complex)
  - vim (powerful, complex)

Editing Files

RUB

- Scripts, configurations, inputs, outputs, logs, ... are usually textfiles.
- Linux provides terminal based editors for textfiles in a huge variaty.
- Famous are:
  - emacs (powerful, complex)
  - vim (powerful, complex)
  - nano (simple, suitable for beginners)

Editing Files

RUB

- Scripts, configurations, inputs, outputs, logs, ... are usually textfiles.
- Linux provides terminal based editors for textfiles in a huge variaty.
- Famous are:
  - emacs (powerful, complex)
  - vim (powerful, complex)
  - nano (simple, suitable for beginners)

Terminal

```
alice@hpc:$ nano
```

Editing Files

RUB

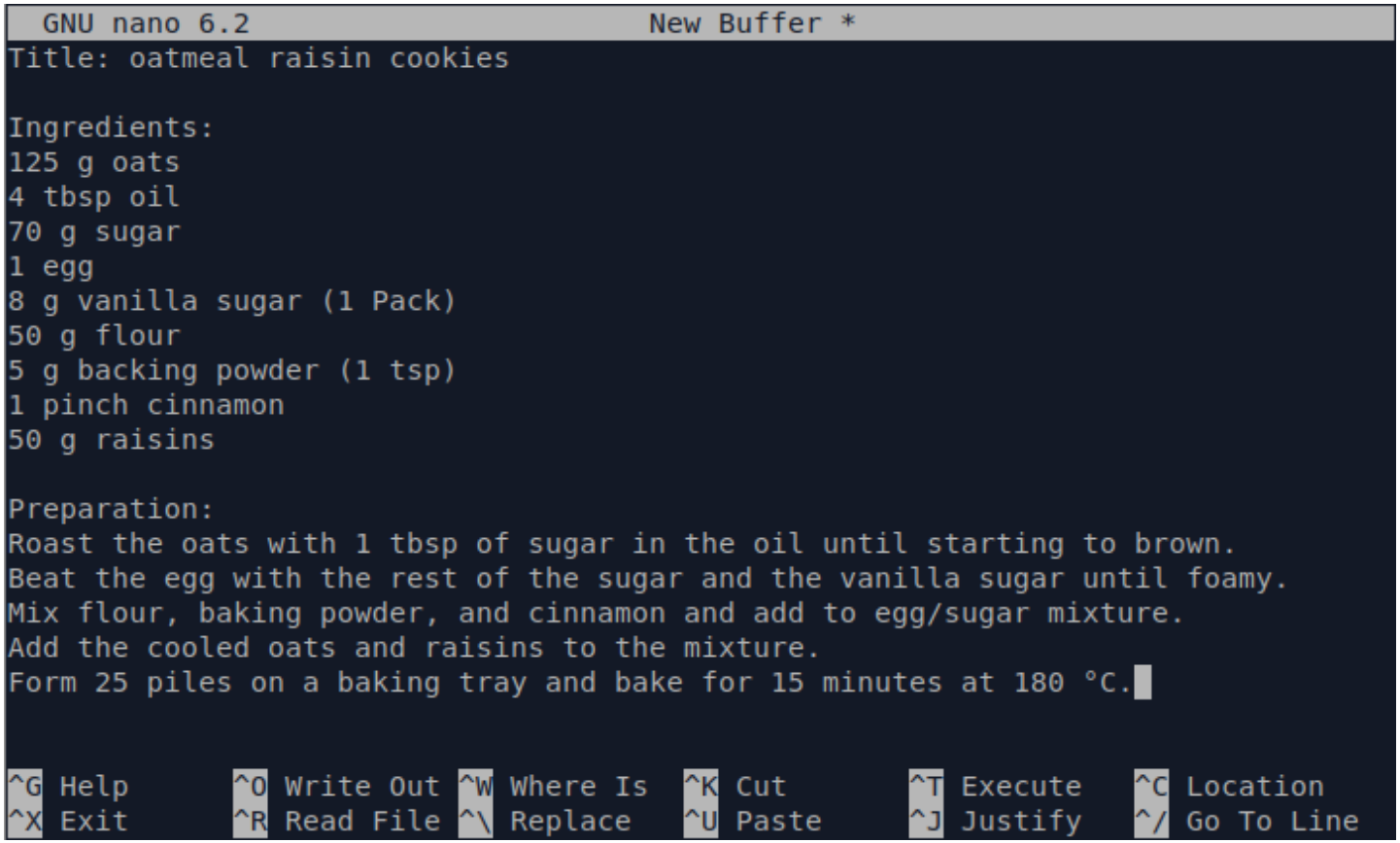
- Scripts, configurations, inputs, outputs, logs, ... are usually textfiles.
- Linux provides terminal based editors for textfiles in a huge variaty.
- Famous are:
  - emacs (powerful, complex)
  - vim (powerful, complex)
  - nano (simple, suitable for beginners)

Terminal

```
alice@hpc:$ nano oatmeal-raisin-cookies.txt
```

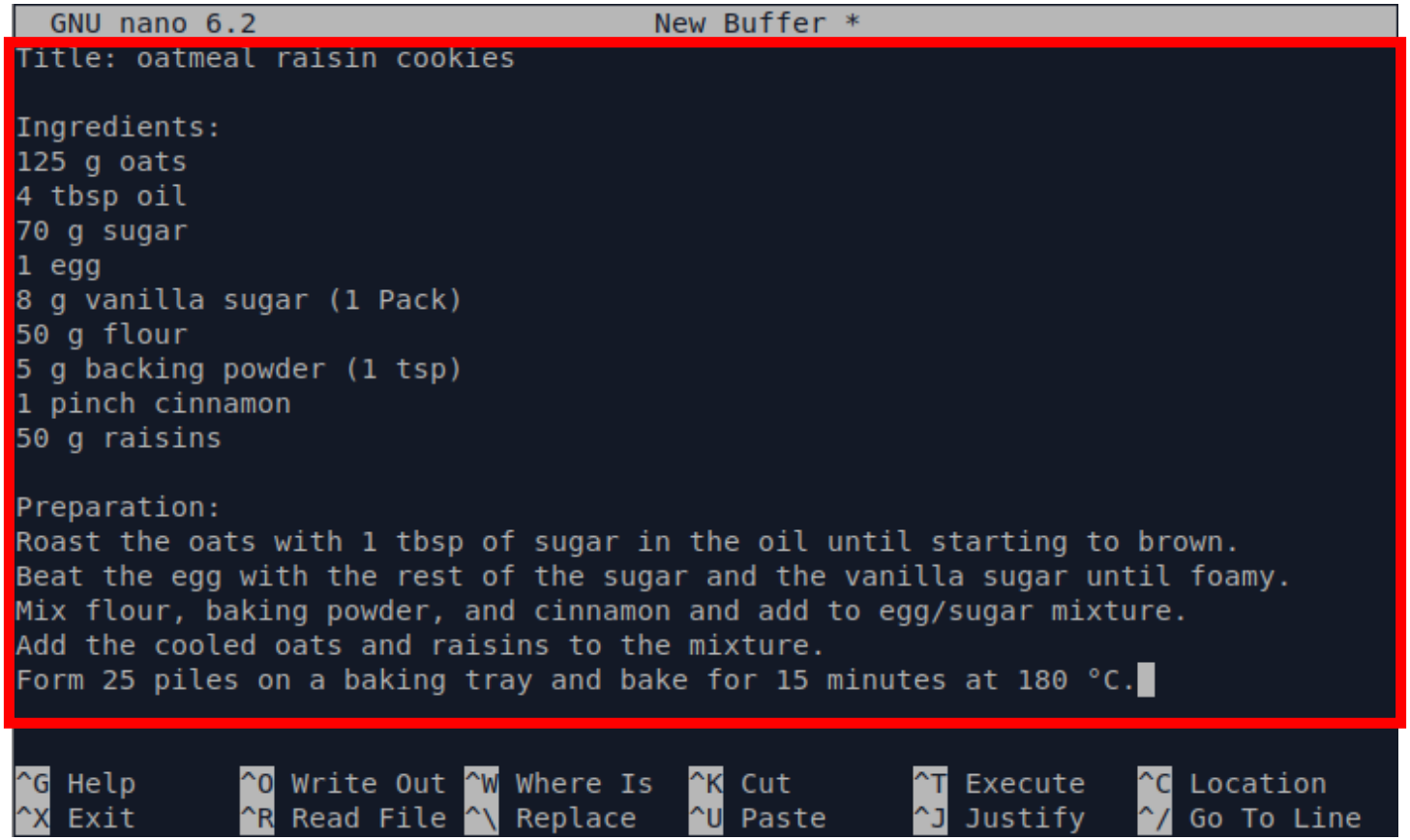
The Nano Editor

RUB



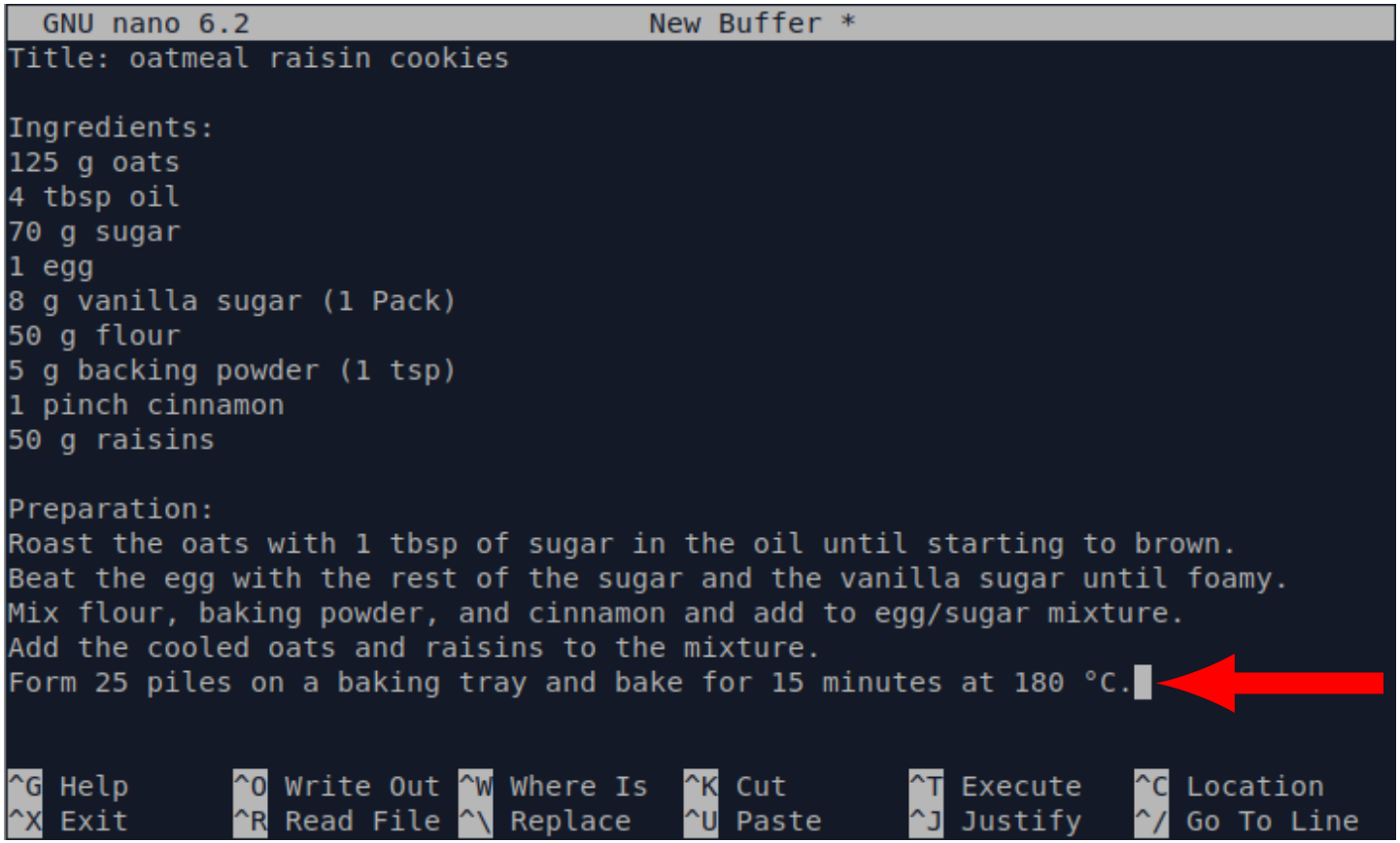
Nano is an easy to learn and use editor.

The Nano Editor



Textfield: Just start typing.

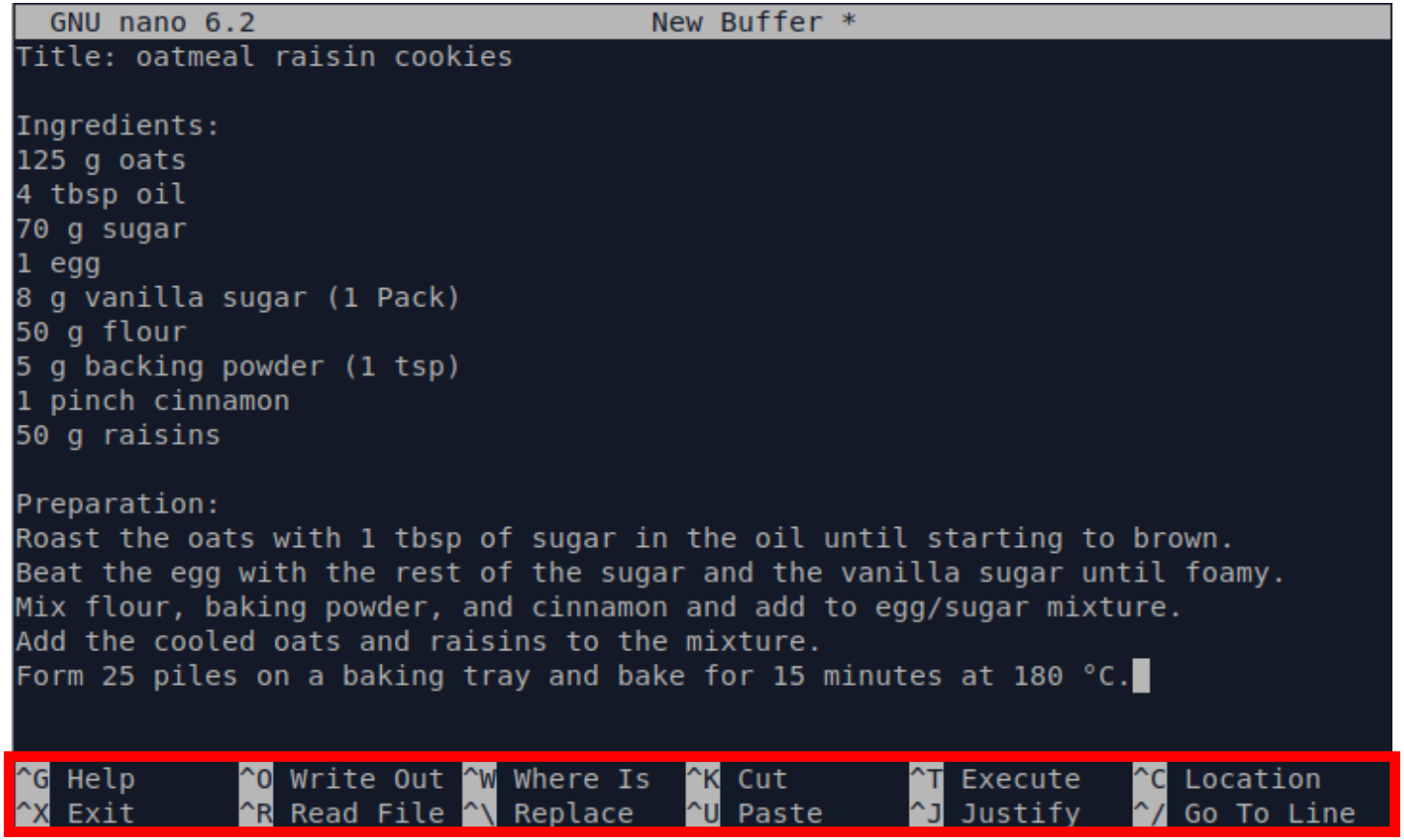
The Nano Editor



Move the cursor with the arrow-keys.

The Nano Editor

RUB



Commands: Perform actions with shortcuts (^ = Ctrl+).

Exercise 4

1. Open `data.txt` with the nano editor
2. Write ten five-digit numbers into your file (One number per line)

Exercise 4

1. Open `data.txt` with the nano editor
2. Write ten five-digit numbers into your file (One number per line)

data.txt

```
34966
85350
62256
62524
82200
59493
97593
92477
40363
93040
```

Workflow and Pipelines

Workflow and Pipelines

RUB

- Linux comes with a huge set of small programs.

Workflow and Pipelines

RUB

- Linux comes with a huge set of small programs.
- Each does only one thing, but it does it well.

Workflow and Pipelines

RUB

- Linux comes with a huge set of small programs.
- Each does only one thing, but it does it well.
- You can take the output of one program and immediately pass it to the next program as input.

Workflow and Pipelines

RUB

- Linux comes with a huge set of small programs.
- Each does only one thing, but it does it well.
- You can take the output of one program and immediately pass it to the next program as input.
- This is done with the pipe operator "|".

Workflow and Pipelines

RUB

- Linux comes with a huge set of small programs.
- Each does only one thing, but it does it well.
- You can take the output of one program and immediately pass it to the next program as input.
- This is done with the pipe operator "|".
- This enables the quick construction of flexible processing pipelines:
  - `command1 | command2 | command3`

Workflow and Pipelines

RUB

- Linux comes with a huge set of small programs.
- Each does only one thing, but it does it well.
- You can take the output of one program and immediately pass it to the next program as input.
- This is done with the pipe operator "|".
- This enables the quick construction of flexible processing pipelines:
  - `command1 | command2 | command3`
- ">" Takes the output of a program and puts it into a file
  - `command1 | command2 | command3 > new_file`
- ">>" Takes the output of a program and appends it to a file
  - `command1 | command2 | command3 >> old_file`

Pipeline Example

RUB

- Example Task: How many numbers between 1 and 1,000,000,000 contain the sequence 65537?

Terminal

```
alice@hpc:$
```

Pipeline Example

- Example Task: How many numbers between 1 and 1,000,000,000 contain the sequence 65537?
- `seq 1000000000` creates a sequence of numbers 1 - 1000000000.

Terminal

```
alice@hpc:$ seq 1000000000
1
2
...
999999999
1000000000
```

Pipeline Example

RUB

- Example Task: How many numbers between 1 and 1,000,000,000 contain the sequence 65537?
- `grep 65537` extracts all lines that contain "65537".

Terminal

```
alice@hpc:$ seq 1000000000 | grep 65537
65537
165537
...
999865537
999965537
```

`grep` can use regular expressions (regex)

Pipeline Example

- Example Task: How many numbers between 1 and 1,000,000,000 contain the sequence 65537?
- `wc -l` counts the lines.

Terminal

```
alice@hpc:$ seq 1000000000 | grep 65537 | wc -l
50000
```

Exercise 5

1. Write the contents of `data.txt` to the terminal with `cat`
2. Sort your numbers with the `sort` command
3. Find the smallest number with `head -n 1`
4. Pipe the smallest number to a new file `limits.txt` with `>`
4. Find the largest number with `tail -n 1`
4. and append it to `limits.txt` with `>>`

Exercise 5

1. Write the contents of `data.txt` to the terminal with `cat`

Terminal

```
alice@hpc:$ cat data.txt
34966
85350
62256
62524
82200
59493
97593
92477
40363
93040
```

Exercise 5

2. Sort your numbers with the `sort` command

Terminal

```
alice@hpc:$ cat data.txt | sort
34966
40363
59493
62256
62524
82200
85350
92477
93040
97593
```

Exercise 5

3. Find the smallest number with `head -n 1`

Terminal

```
alice@hpc:$ cat data.txt | sort | head -n 1
34966
```

Exercise 5

4. Pipe the smallest number to a new file `limits.txt` with >

Terminal

```
alice@hpc:$ cat data.txt | sort | head -n 1 > limits.txt
```

Exercise 5

4. Find the largest number with `tail -n 1`

Terminal

```
alice@hpc:$ cat data.txt | sort | tail -n 1
93040
```

Exercise 5

4. and append it to `limits.txt` with `>>`

Terminal

```
alice@hpc:$ cat data.txt | sort | tail -n 1 >> limits.txt
```

Exercise 5

RUB

Terminal

```
alice@hpc:$ cat limits.txt
34966
97593
```

Automation and Scripting

Variables

It is possible to store text snippets in variables for later use.

Terminal

```
alice@hpc:$ long_text="All your base are belong to us."
```

Variables

Variables can be accessed with `${variablename}`.

Terminal

```
alice@hpc:$ long_text="All your base are belong to us."
alice@hpc:$ echo ${long_text}
All your base are belong to us.
```

Variables

Commands can also be stored in variables.

Terminal

```
alice@hpc:$ long_text="All your base are belong to us."
alice@hpc:$ echo ${long_text}
All your base are belong to us.
alice@hpc:$ program="echo"
```

Variables

Commands can also be stored in variables.

Terminal

```
alice@hpc:$ long_text="All your base are belong to us."
alice@hpc:$ echo ${long_text}
All your base are belong to us.
alice@hpc:$ program="echo"
alice@hpc:$ ${program} ${long_text}
All your base are belong to us.
```

Variables

Variables can again be stored in variables.

Terminal

```
alice@hpc:$ long_text="All your base are belong to us."
alice@hpc:$ echo ${long_text}
All your base are belong to us.
alice@hpc:$ program="echo"
alice@hpc:$ ${program} ${long_text}
All your base are belong to us.
alice@hpc:$ command="${program} ${long_text}"
```

Variables

Variables can again be stored in variables.

Terminal

```
alice@hpc:$ long_text="All your base are belong to us."
alice@hpc:$ echo ${long_text}
All your base are belong to us.
alice@hpc:$ program="echo"
alice@hpc:$ ${program} ${long_text}
All your base are belong to us.
alice@hpc:$ command="${program} ${long_text}"
alice@hpc:$ ${command}
All your base are belong to us.
```

Numerical Variables and Calculations

RUB

`bc` (basic calculator) can be used to handle simple calculations

Terminal

```
alice@hpc:$ echo "(2*3+11-7)^3/5" | bc
200
```

Numerical Variables and Calculations

RUB

`bc` (basic calculator) can be used to handle simple calculations

Terminal

```
alice@hpc:$ echo "(2*3+11-7)^3/5" | bc
200
alice@hpc:$ echo "(2*3+11-5)/7" | bc
1
```

Numerical Variables and Calculations

the `-l` flag is required for floating point arithmetic

Terminal

```
alice@hpc:$ echo "(2*3+11-7)^3/5" | bc
200
alice@hpc:$ echo "(2*3+11-5)/7" | bc
1
alice@hpc:$ echo "(2*3+11-5)/7" | bc -l
1.71428571428571428571
```

Numerical Variables and Calculations

RUB

Results can be stored in variables with $()

Terminal

```
alice@hpc:$ echo "(2*3+11-7)^3/5" | bc
200
alice@hpc:$ echo "(2*3+11-5)/7" | bc
1
alice@hpc:$ echo "(2*3+11-5)/7" | bc -l
1.71428571428571428571
alice@hpc:$ pi=$(echo "22/7" | bc -l)
```

Numerical Variables and Calculations

RUB

Results can be stored in variables with `$()`, and used again

Terminal

```
alice@hpc:$ echo "(2*3+11-7)^3/5" | bc
200
alice@hpc:$ echo "(2*3+11-5)/7" | bc
1
alice@hpc:$ echo "(2*3+11-5)/7" | bc -l
1.71428571428571428571
alice@hpc:$ pi=$(echo "22/7" | bc -l)
alice@hpc:$ echo "${pi}*2" | bc -l
6.28571428571428571428
```

Writing Scripts

RUB

myscript.sh

```
#!/bin/bash
```

Select bash as language for this script

Writing Scripts

RUB

myscript.sh

```
#!/bin/bash
# write a few infos to variables
```

Comments start with "#" and are ignored

Writing Scripts

RUB

myscript.sh

```
#!/bin/bash
# write a few infos to variables
user=$(whoami)
host=$(hostname)
timestamp=$(date +"%s")
```

"$()" executes the command inside and assigns it to the variable

Writing Scripts

RUB

myscript.sh

```
#!/bin/bash
# write a few infos to variables
user=$(whoami)
host=$(hostname)
timestamp=$(date +"%s")
# combine into string
string="executed by ${user} on ${host} at ${timestamp}"
```

Store a string in a variable

Writing Scripts

RUB

myscript.sh

```
#!/bin/bash
# write a few infos to variables
user=$(whoami)
host=$(hostname)
timestamp=$(date +"%s")
# combine into string
string="executed by ${user} on ${host} at ${timestamp}"
echo ${string}
```

Output the string to the terminal

Running Scripts

RUB

Terminal

```
alice@hpc:$ bash ./myscript.sh
```

Run a script by calling the interpreter "`bash`",
and giving it the script path as argument.

Running Scripts

RUB

Terminal

```
alice@hpc:$ bash ./myscript.sh
executed by alice on hpc at We 31.  Jan 09:51:07 CEST 2024
```

Run a script by calling the interpreter "`bash`",
and giving it the script path as argument.

Running Scripts

RUB

Terminal

```
alice@hpc:$ ls -l ./myscript.sh
-rwxr-xr-- 1 alice phys 575 Jan 31 06:52 myscript.sh
```

If the script is marked executable the interpreter can be omitted.
The interpreter will be taken from the first line of the script.

Running Scripts

RUB

Terminal

```
alice@hpc:$ ls -l ./myscript.sh
-rwxr-xr-- 1 alice phys 575 Jan 31 06:52 myscript.sh
alice@hpc:$ ./myscript.sh
```

If the script is marked executable the interpreter can be omitted.
The interpreter will be taken from the first line of the script.

Running Scripts

RUB

Terminal

```
alice@hpc:$ ls -l ./myscript.sh
-rwxr-xr-- 1 alice phys 575 Jan 31 06:52 myscript.sh
alice@hpc:$ ./myscript.sh
executed by alice on hpc at We 31.  Jan 09:55:07 CEST 2024
```

If the script is marked executable the interpreter can be omitted.
The interpreter will be taken from the first line of the script.

Exercise 6

1. Write a script that...
   - stores the min and max value from `data.txt` in variables
   - computes the sum and difference of the min and max values and stores them in variables
   - writes out the values of the sum and difference to the terminal
   - computes and then writes out one, two, and three times the minimal value.
2. Run the script.

Exercise 6

1. Write a script that...

myscript.sh

```
#!/bin/bash
```

Exercise 6

1. Write a script that...
   - stores the min and max value from `data.txt` in variables

myscript.sh

```
#!/bin/bash
minval=$(cat data.txt | sort | head -n 1)
maxval=$(cat data.txt | sort | tail -n 1)
```

Exercise 6

1. Write a script that...
   - computes the sum and difference of the min and max values and stores them in variables

myscript.sh

```
#!/bin/bash
minval=$(cat data.txt | sort | head -n 1)
maxval=$(cat data.txt | sort | tail -n 1)
sumstr="${maxval} + ${minval}"
difstr="${maxval} - ${minval}"
```

Exercise 6

RUB

1. Write a script that...
   - computes the sum and difference of the min and max values and stores them in variables

myscript.sh

```
#!/bin/bash
minval=$(cat data.txt | sort | head -n 1)
maxval=$(cat data.txt | sort | tail -n 1)
sumstr="${maxval} + ${minval}"
difstr="${maxval} - ${minval}"
sum=$(echo "${sumstr}" | bc)
dif=$(echo "${difstr}" | bc)
```

Exercise 6

1. Write a script that...
   - writes out the values of the sum and difference to the terminal

myscript.sh

```
#!/bin/bash
minval=$(cat data.txt | sort | head -n 1)
maxval=$(cat data.txt | sort | tail -n 1)
sumstr="${maxval} + ${minval}"
difstr="${maxval} - ${minval}"
sum=$(echo "${sumstr}" | bc)
dif=$(echo "${difstr}" | bc)
echo "${sumstr} = ${sum}"
echo "${difstr} = ${dif}"
```

Exercise 6

RUB

1. Write a script that...
   - computes and then writes out one, two, and three times the minimal value.

myscript.sh

```
#!/bin/bash
minval=$(cat data.txt | sort | head -n 1)
maxval=$(cat data.txt | sort | tail -n 1)
sumstr="${maxval} + ${minval}"
difstr="${maxval} - ${minval}"
sum=$(echo "${sumstr}" | bc)
dif=$(echo "${difstr}" | bc)
echo "${sumstr} = ${sum}"
echo "${difstr} = ${dif}"
value=$(echo "1*${minval}" | bc)
echo "1*${minval} = ${value}"
value=$(echo "2*${minval}" | bc)
echo "2*${minval} = ${value}"
value=$(echo "3*${minval}" | bc)
echo "3*${minval} = ${value}"
```

Exercise 6

2. Run the script.

Terminal

```
alice@hpc:$
```

Exercise 6

2. Run the script.

Terminal

```
alice@hpc:$ ./script.sh
```

Exercise 6

2. Run the script.

Terminal

```
alice@hpc:$ ./script.sh
97593 + 34966 = 132559
97593 - 34966 = 62627
1*34966 = 34966
2*34966 = 69932
3*34966 = 104898
```

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
value=$(echo "1*${minval}" | bc)
echo "1*${minval} = ${value}"
```

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
value=$(echo "1*${minval}" | bc)
echo "1*${minval} = ${value}"
value=$(echo "2*${minval}" | bc)
echo "2*${minval} = ${value}"
```

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
value=$(echo "1*${minval}" | bc)
echo "1*${minval} = ${value}"
value=$(echo "2*${minval}" | bc)
echo "2*${minval} = ${value}"
value=$(echo "3*${minval}" | bc)
echo "3*${minval} = ${value}"
```

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
value=$(echo "1*${minval}" | bc)
echo "1*${minval} = ${value}"
value=$(echo "2*${minval}" | bc)
echo "2*${minval} = ${value}"
value=$(echo "3*${minval}" | bc)
echo "3*${minval} = ${value}"
```

Repetitive tasks often can be made easier:

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
value=$(echo "1*${minval}" | bc)
echo "1*${minval} = ${value}"
value=$(echo "2*${minval}" | bc)
echo "2*${minval} = ${value}"
value=$(echo "3*${minval}" | bc)
echo "3*${minval} = ${value}"
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
i=1
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes

Basic Loop Structure

RUB

Some tasks repeat multiple times with only slight variations:

Terminal

```
i=1
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
i=2
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
i=1
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
i=2
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
i=3
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
for
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes
3. use a for-loop structure to automatically adjust the value

Basic Loop Structure

RUB

Some tasks repeat multiple times with only slight variations:

Terminal

```
for i
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes
3. use a for-loop structure to automatically adjust the value

Basic Loop Structure

RUB

Some tasks repeat multiple times with only slight variations:

Terminal

```
for i in
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes
3. use a for-loop structure to automatically adjust the value

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
for i in 1 2 3
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes
3. use a for-loop structure to automatically adjust the value

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
for i in 1 2 3
do


done
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes
3. use a for-loop structure to automatically adjust the value

Basic Loop Structure

RUB

Some tasks repeat multiple times with only slight variations:

Terminal

```
for i in 1 2 3
do
    value=$(echo "${i}*${minval}" | bc)
    echo "${i}*${minval} = ${value}"
done
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes
3. use a for-loop structure to automatically adjust the value

Basic Loop Structure

Some tasks repeat multiple times with only slight variations:

Terminal

```
for i in $(seq 3)
do
    value=$(echo "${i}*${minval}" | bc)
    echo "${i}*${minval} = ${value}"
done
```

Repetitive tasks often can be made easier:

1. isolate the "change" from one repetition to another
2. introduce a variable that incorporates all that changes
3. use a for-loop structure to automatically adjust the value
4. use the seq command to abbreviate the value list

Basic Loop Structure

RUB

Terminal

```
for i in <list>
do
...
done
```

1. `i` behaves like a normal variable in bash

Basic Loop Structure

RUB

Terminal

```
for i in <list>
do
...
done
```

1. `i` behaves like a normal variable in bash
2. `list` can be a list of numbers, files, words, ...

Exercise 7

1. Adjust your script such that...
   - the repetetive multiplication of the minval is incorporated into a loop.
   - the multiplication goes up to 10 times the minimal value.
2. Run the script.

Exercise 7

1. Adjust your script such that...
   - the repetetive multiplication of the minval is incorporated into a loop.

myscript.sh

```
#!/bin/bash
...
value=$(echo "1*${minval}" | bc)
echo "1*${minval} = ${value}"
value=$(echo "2*${minval}" | bc)
echo "2*${minval} = ${value}"
value=$(echo "3*${minval}" | bc)
echo "3*${minval} = ${value}"
```

Exercise 7

1. Adjust your script such that...
   - the repetetive multiplication of the minval is incorporated into a loop.

myscript.sh

```
#!/bin/bash
...
i=1
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
i=2
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
i=3
value=$(echo "${i}*${minval}" | bc)
echo "${i}*${minval} = ${value}"
```

Exercise 7

1. Adjust your script such that...
   - the repetetive multiplication of the minval is incorporated into a loop.

myscript.sh

```
#!/bin/bash
...
for i in 1 2 3
do
    value=$(echo "${i}*${minval}" | bc)
    echo "${i}*${minval} = ${value}"
done
```

Exercise 7

1. Adjust your script such that...
   - the repetetive multiplication of the minval is incorporated into a loop.

myscript.sh

```
#!/bin/bash
...
for i in $(seq 3)
do
    value=$(echo "${i}*${minval}" | bc)
    echo "${i}*${minval} = ${value}"
done
```

Exercise 7

1. Adjust your script such that...
   - the multiplication goes up to 10 times the minimal value.

myscript.sh

```
#!/bin/bash
...
for i in $(seq 10)
do
    value=$(echo "${i}*${minval}" | bc)
    echo "${i}*${minval} = ${value}"
done
```

Exercise 7

2. Run the script.

Terminal

```
alice@hpc:$
```

Exercise 7

2. Run the script.

Terminal

```
alice@hpc:$ ./script.sh
```

Exercise 7

2. Run the script.

Terminal

```
alice@hpc:$ ./script.sh
97593 + 34966 = 132559
97593 - 34966 = 62627
1*34966 = 34966
2*34966 = 69932
...
10*34966 = 349660
```

Environment variables

Variable Scopes

RUB

outer.sh

```
#!/bin/bash
user=$(whoami)
echo "User in outer script: ${user}"
./inner.sh
```

inner.sh

```
#!/bin/bash
echo "User in inner script: ${user}"
```

The variable `user` is defined in the outer script,
but accessed in the inner and outer one.

Variable Scopes

RUB

Terminal

```
alice@hpc:$ ./outer.sh
User in outer script: alice
User in inner script:
```

The value of variables is not inherited
to sub-scripts, or sub-programs

Variable Scopes

RUB

outer.sh

```
#!/bin/bash
export user=$(whoami)
echo "User in outer script: ${user}"
./inner.sh
```

inner.sh

```
#!/bin/bash
echo "User in inner script: ${user}"
```

Adding `export` to the variable assignment
makes it globaly available $\Rightarrow$ Environment variable

Variable Scopes

RUB

Terminal

```
alice@hpc:$ ./outer.sh
User in outer script: alice
User in inner script: alice
```

Adding `export` to the variable assignment
makes it globaly available ⇒ Environment variable

Predefined Environment Variables

RUB

Terminal

```
alice@hpc:$ env
```

There are predefined environment variables. (Full list with `env`)

Predefined Environment Variables

RUB

Terminal

```
alice@hpc:$ echo ${HOME}
/home/alice/
```

e.g. `${HOME}` contains the path to a users home directory

Predefined Environment Variables

RUB

Terminal

```
alice@hpc:$ echo ${PATH}
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

`${PATH}` contains a list of paths
where program executables are searched for

Predefined Environment Variables

RUB

Terminal

```
alice@hpc:$ echo ${PATH}
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
alice@hpc:$ which gcc
/usr/bin/gcc
```

`${PATH}` contains a list of paths
where program executables are searched for

Predefined Environment Variables

RUB

Terminal

```
alice@hpc:$ export PATH=${HOME}/bin:${PATH}
alice@hpc:$ echo ${PATH}
/home/alice/bin/:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Modifying `${PATH}` allows to utilize custom versions of software

Predefined Environment Variables

RUB

Terminal

```
alice@hpc:$ export PATH=${HOME}/bin:${PATH}
alice@hpc:$ echo ${PATH}
/home/alice/bin/:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
alice@hpc:$ which gcc
/home/alice/bin/gcc
```

Modifying `${PATH}` allows to utilize custom versions of software

Monitoring System Resources

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

Execute "`top`" to start. Quit with Ctrl+c.

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
34600	fuhl	20	0	8716	1032	944	R	100.0	0.0	0:15.21	yes
34603	fuhl	20	0	8716	1004	916	R	100.0	0.0	0:15.21	yes
34604	fuhl	20	0	8716	1008	916	R	100.0	0.0	0:15.22	yes
34599	fuhl	20	0	8716	1024	936	R	99.7	0.0	0:15.21	yes
34601	fuhl	20	0	8716	1008	916	R	99.7	0.0	0:15.21	yes
34602	fuhl	20	0	8716	1056	964	R	99.7	0.0	0:15.21	yes
29175	fuhl	20	0	2997864	447460	120320	S	3.3	2.8	4:58.28	Isolate+
1816	fuhl	20	0	1124.1g	193868	129364	S	2.3	1.2	4:21.85	element+
2336	fuhl	20	0	1129.9g	330224	116724	S	2.3	2.1	22:59.17	ucware-+
10140	fuhl	20	0	4269040	501736	258080	S	2.3	3.2	17:29.36	firefox
2306	fuhl	20	0	1135.9g	297668	99408	S	1.3	1.9	3:49.92	element+
1090	root	20	0	2313152	119156	73032	S	1.0	0.8	14:21.24	Xorg
2132	fuhl	20	0	32.3g	74320	62972	S	0.7	0.5	2:53.10	element+
10271	fuhl	20	0	2473236	128716	99116	S	0.7	0.8	0:03.99	Privile+
10384	fuhl	20	0	345136	56948	45148	S	0.7	0.4	2:39.80	Utility+
14	root	20	0	0	0	0	I	0.3	0.0	0:24.86	rcu_sch+
976	root	20	0	276576	11140	10116	S	0.3	0.1	0:35.15	thermald

System resources

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

Load average for past one, five, and fifteen minutes

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

CPU-time (us→user, sy→system, id→idle)

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

Memory load

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

Swap load (try to avoid any swap load)

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
34600	**fuhl**	**20**	**0**	**8716**	**1032**	**944**	**R**	**100.0**	**0.0**	**0:15.21**	**yes**
34603	**fuhl**	**20**	**0**	**8716**	**1004**	**916**	**R**	**100.0**	**0.0**	**0:15.21**	**yes**
34604	**fuhl**	**20**	**0**	**8716**	**1008**	**916**	**R**	**100.0**	**0.0**	**0:15.22**	**yes**
34599	**fuhl**	**20**	**0**	**8716**	**1024**	**936**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
34601	**fuhl**	**20**	**0**	**8716**	**1008**	**916**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
34602	**fuhl**	**20**	**0**	**8716**	**1056**	**964**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
29175	fuhl	20	0	2997864	447460	120320	S	3.3	2.8	4:58.28	Isolate+
1816	fuhl	20	0	1124.1g	193868	129364	S	2.3	1.2	4:21.85	element+
2336	fuhl	20	0	1129.9g	330224	116724	S	2.3	2.1	22:59.17	ucware-+
10140	fuhl	20	0	4269040	501736	258080	S	2.3	3.2	17:29.36	firefox
2306	fuhl	20	0	1135.9g	297668	99408	S	1.3	1.9	3:49.92	element+
1090	root	20	0	2313152	119156	73032	S	1.0	0.8	14:21.24	Xorg
2132	fuhl	20	0	32.3g	74320	62972	S	0.7	0.5	2:53.10	element+
10271	fuhl	20	0	2473236	128716	99116	S	0.7	0.8	0:03.99	Privile+
10384	fuhl	20	0	345136	56948	45148	S	0.7	0.4	2:39.80	Utility+
14	root	20	0	0	0	0	I	0.3	0.0	0:24.86	rcu_sch+
976	root	20	0	276576	11140	10116	S	0.3	0.1	0:35.15	thermald

List of running processes

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

Process ID

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

Process owner

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

CPU load from this process

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
34600	**fuhl**	**20**	**0**	**8716**	**1032**	**944**	**R**	**100.0**	**0.0**	**0:15.21**	**yes**
34603	**fuhl**	**20**	**0**	**8716**	**1004**	**916**	**R**	**100.0**	**0.0**	**0:15.21**	**yes**
34604	**fuhl**	**20**	**0**	**8716**	**1008**	**916**	**R**	**100.0**	**0.0**	**0:15.22**	**yes**
34599	**fuhl**	**20**	**0**	**8716**	**1024**	**936**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
34601	**fuhl**	**20**	**0**	**8716**	**1008**	**916**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
34602	**fuhl**	**20**	**0**	**8716**	**1056**	**964**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
29175	fuhl	20	0	2997864	447460	120320	S	3.3	2.8	4:58.28	Isolate+
1816	fuhl	20	0	1124.1g	193868	129364	S	2.3	1.2	4:21.85	element+
2336	fuhl	20	0	1129.9g	330224	116724	S	2.3	2.1	22:59.17	ucware-+
10140	fuhl	20	0	4269040	501736	258080	S	2.3	3.2	17:29.36	firefox
2306	fuhl	20	0	1135.9g	297668	99408	S	1.3	1.9	3:49.92	element+
1090	root	20	0	2313152	119156	73032	S	1.0	0.8	14:21.24	Xorg
2132	fuhl	20	0	32.3g	74320	62972	S	0.7	0.5	2:53.10	element+
10271	fuhl	20	0	2473236	128716	99116	S	0.7	0.8	0:03.99	Privile+
10384	fuhl	20	0	345136	56948	45148	S	0.7	0.4	2:39.80	Utility+
14	root	20	0	0	0	0	I	0.3	0.0	0:24.86	rcu_sch+
976	root	20	0	276576	11140	10116	S	0.3	0.1	0:35.15	thermald

Memory usage of this process

top — Table of Processes

RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
34600	**fuhl**	**20**	**0**	**8716**	**1032**	**944**	**R**	**100.0**	**0.0**	**0:15.21**	**yes**
34603	**fuhl**	**20**	**0**	**8716**	**1004**	**916**	**R**	**100.0**	**0.0**	**0:15.21**	**yes**
34604	**fuhl**	**20**	**0**	**8716**	**1008**	**916**	**R**	**100.0**	**0.0**	**0:15.22**	**yes**
34599	**fuhl**	**20**	**0**	**8716**	**1024**	**936**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
34601	**fuhl**	**20**	**0**	**8716**	**1008**	**916**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
34602	**fuhl**	**20**	**0**	**8716**	**1056**	**964**	**R**	**99.7**	**0.0**	**0:15.21**	**yes**
29175	fuhl	20	0	2997864	447460	120320	S	3.3	2.8	4:58.28	Isolate+
1816	fuhl	20	0	1124.1g	193868	129364	S	2.3	1.2	4:21.85	element+
2336	fuhl	20	0	1129.9g	330224	116724	S	2.3	2.1	22:59.17	ucware-+
10140	fuhl	20	0	4269040	501736	258080	S	2.3	3.2	17:29.36	firefox
2306	fuhl	20	0	1135.9g	297668	99408	S	1.3	1.9	3:49.92	element+
1090	root	20	0	2313152	119156	73032	S	1.0	0.8	14:21.24	Xorg
2132	fuhl	20	0	32.3g	74320	62972	S	0.7	0.5	2:53.10	element+
10271	fuhl	20	0	2473236	128716	99116	S	0.7	0.8	0:03.99	Privile+
10384	fuhl	20	0	345136	56948	45148	S	0.7	0.4	2:39.80	Utility+
14	root	20	0	0	0	0	I	0.3	0.0	0:24.86	rcu_sch+
976	root	20	0	276576	11140	10116	S	0.3	0.1	0:35.15	thermald

Time since process started

top — Table of Processes

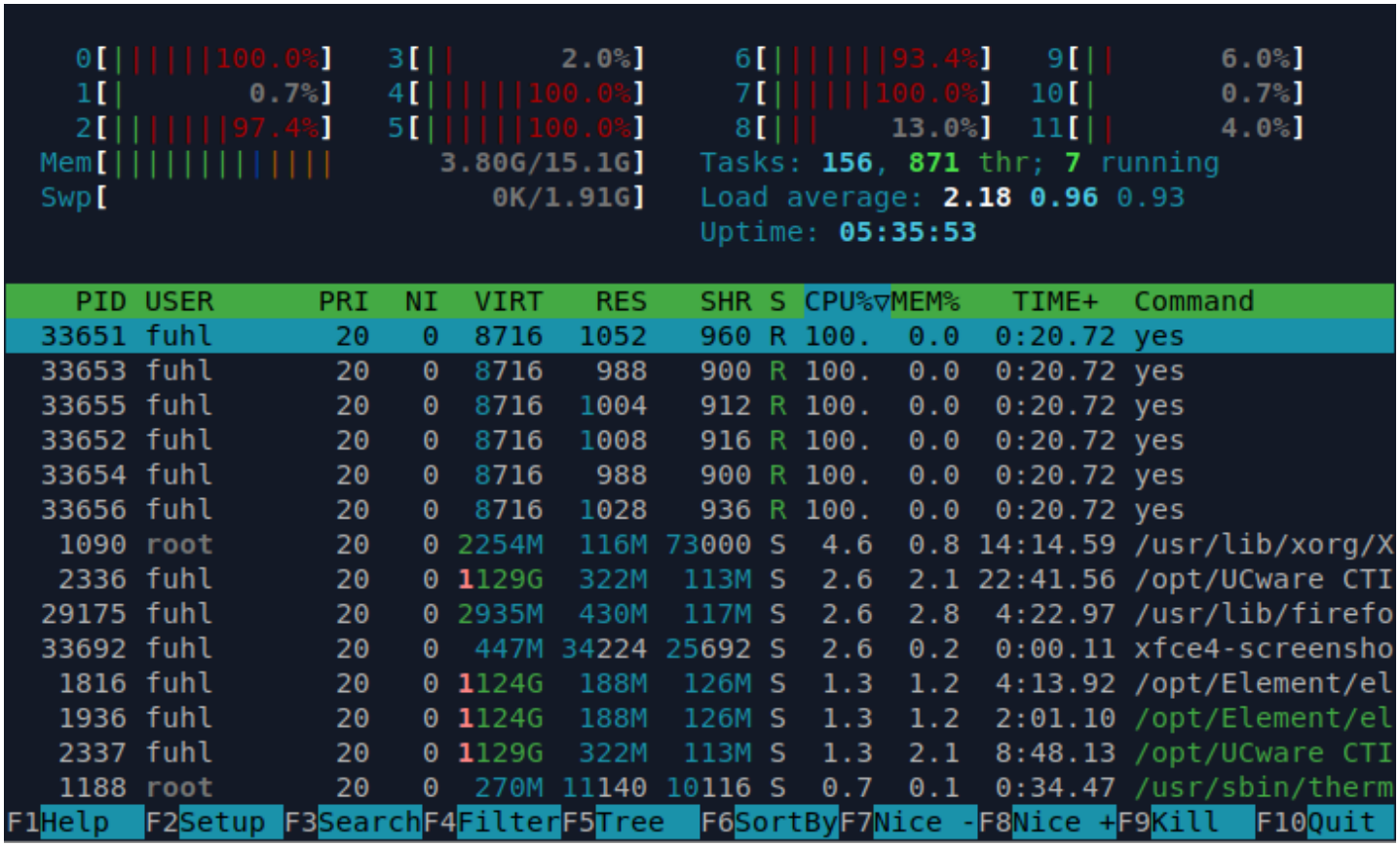
RUB

```
top - 09:12:33 up  5:40,  1 user,  load average: 1.83, 1.27, 1.10
Tasks: 352 total,   7 running, 345 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us, 47.5 sy,  0.0 ni, 48.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15421.4 total,   9117.5 free,   2767.8 used,   3536.1 buff/cache
MiB Swap:   1956.0 total,   1956.0 free,      0.0 used.  11116.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  34600 fuhl      20   0    8716   1032    944 R 100.0   0.0   0:15.21 yes
  34603 fuhl      20   0    8716   1004    916 R 100.0   0.0   0:15.21 yes
  34604 fuhl      20   0    8716   1008    916 R 100.0   0.0   0:15.22 yes
  34599 fuhl      20   0    8716   1024    936 R  99.7   0.0   0:15.21 yes
  34601 fuhl      20   0    8716   1008    916 R  99.7   0.0   0:15.21 yes
  34602 fuhl      20   0    8716   1056    964 R  99.7   0.0   0:15.21 yes
  29175 fuhl      20   0 2997864 447460 120320 S   3.3   2.8   4:58.28 Isolate+
   1816 fuhl      20   0 1124.1g 193868 129364 S   2.3   1.2   4:21.85 element+
   2336 fuhl      20   0 1129.9g 330224 116724 S   2.3   2.1  22:59.17 ucware-+
  10140 fuhl      20   0 4269040 501736 258080 S   2.3   3.2  17:29.36 firefox
   2306 fuhl      20   0 1135.9g 297668  99408 S   1.3   1.9   3:49.92 element+
   1090 root      20   0 2313152 119156  73032 S   1.0   0.8  14:21.24 Xorg
   2132 fuhl      20   0   32.3g  74320  62972 S   0.7   0.5   2:53.10 element+
  10271 fuhl      20   0 2473236 128716  99116 S   0.7   0.8   0:03.99 Privile+
  10384 fuhl      20   0  345136  56948  45148 S   0.7   0.4   2:39.80 Utility+
     14 root      20   0       0      0      0 I   0.3   0.0   0:24.86 rcu_sch+
    976 root      20   0  276576  11140  10116 S   0.3   0.1   0:35.15 thermald
```

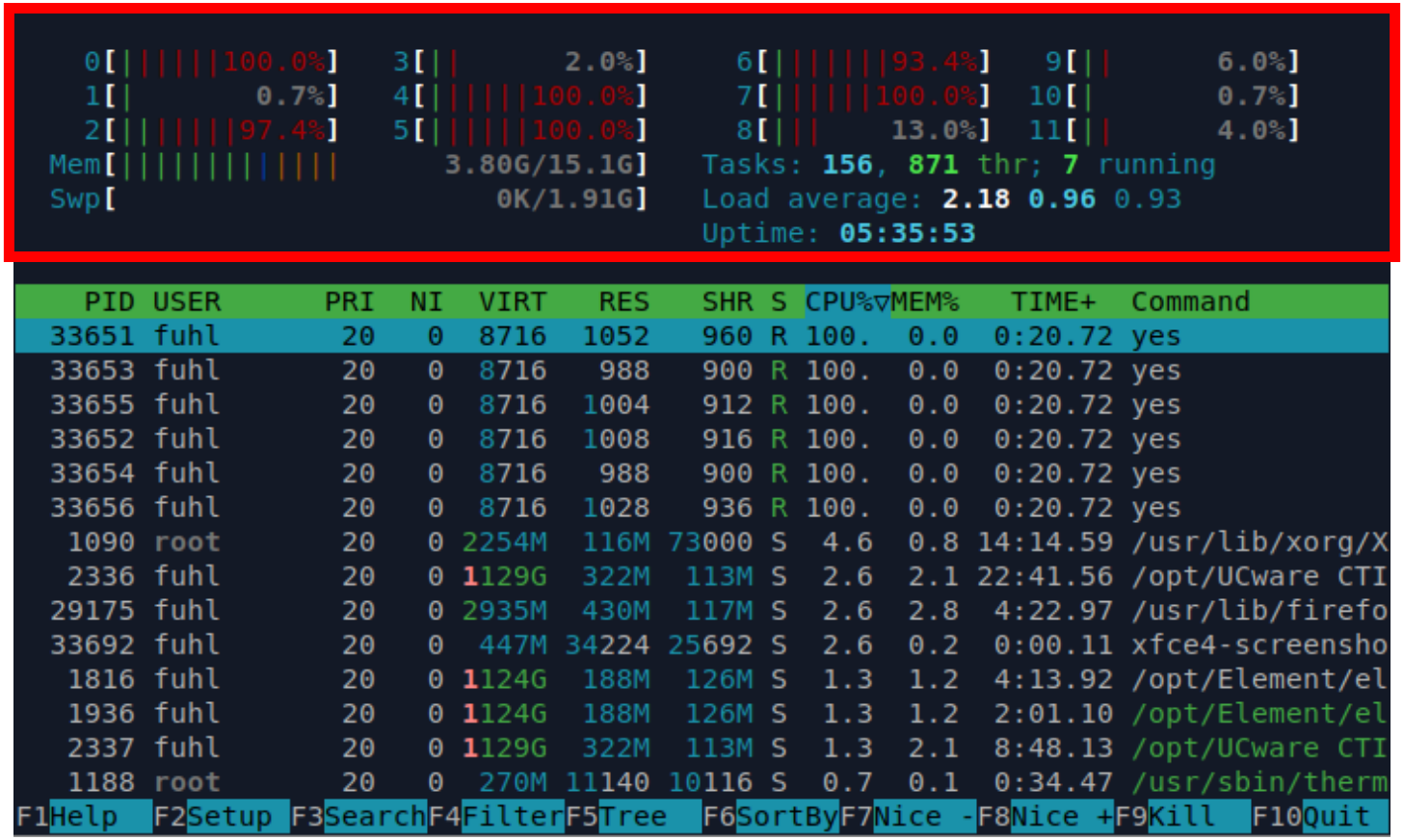
Command that started the process

htop — An Alternative to top



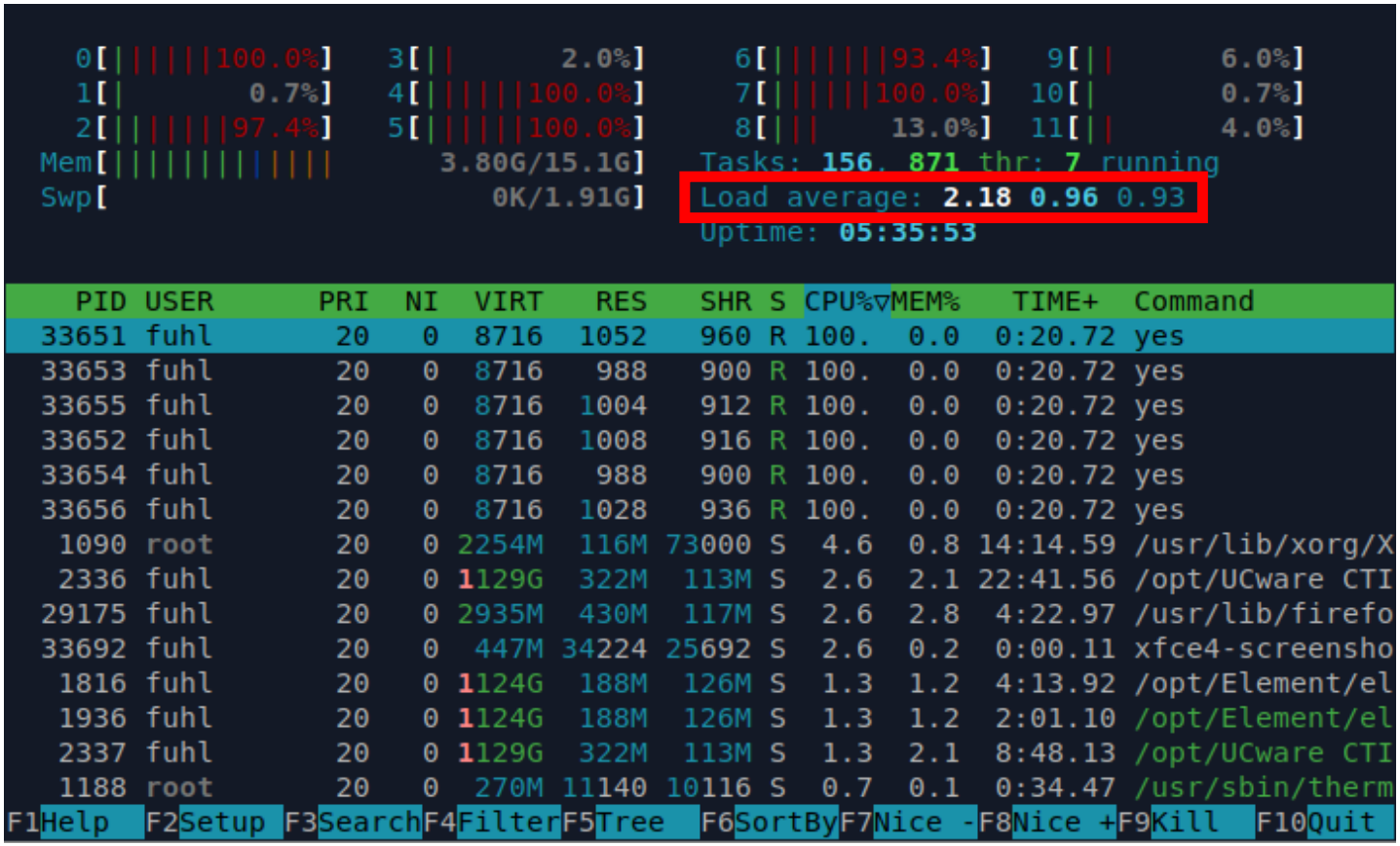
Execute "`htop`" to start. Quit with Ctrl+c.

htop — An Alternative to top



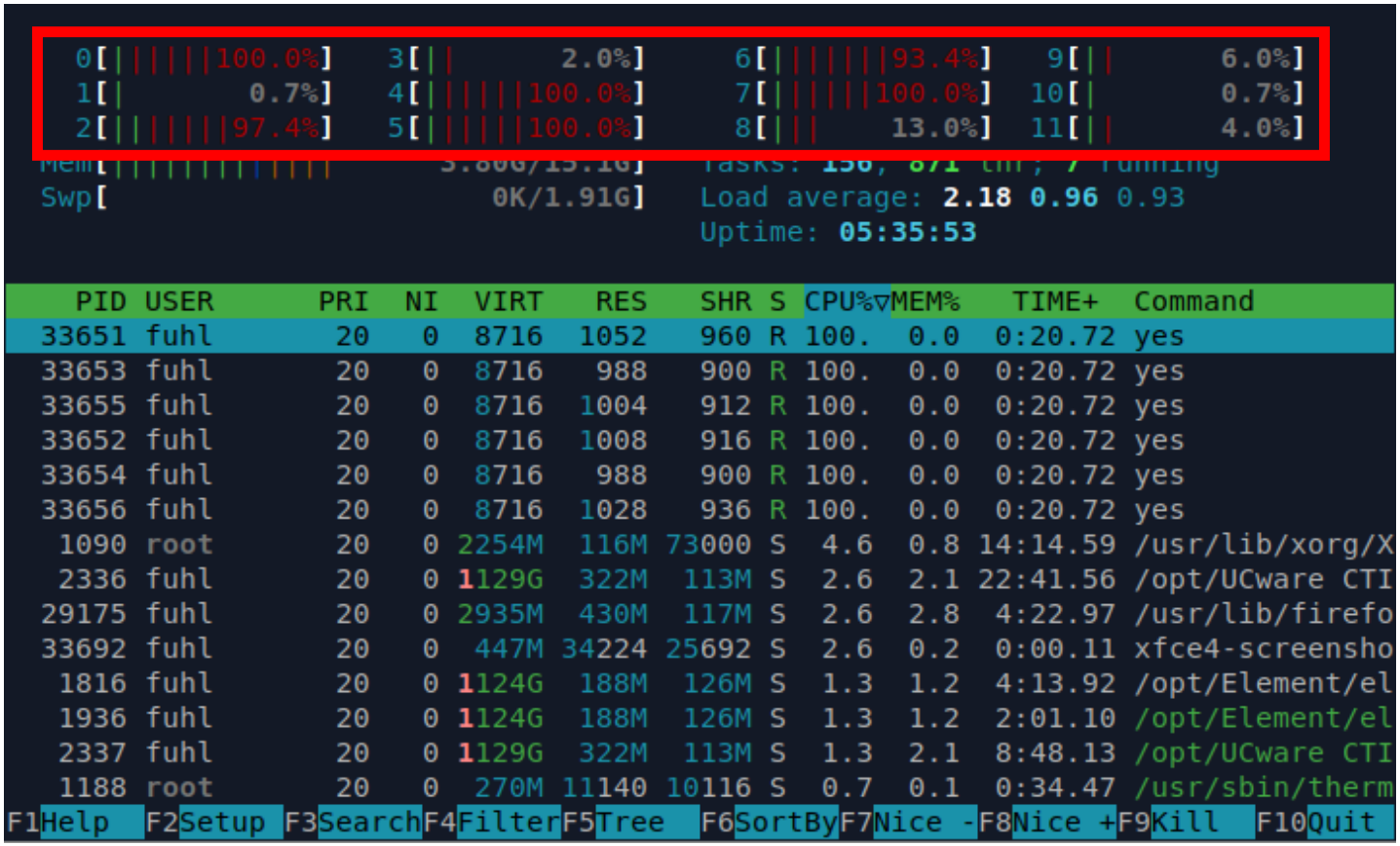
System resources

htop — An Alternative to top



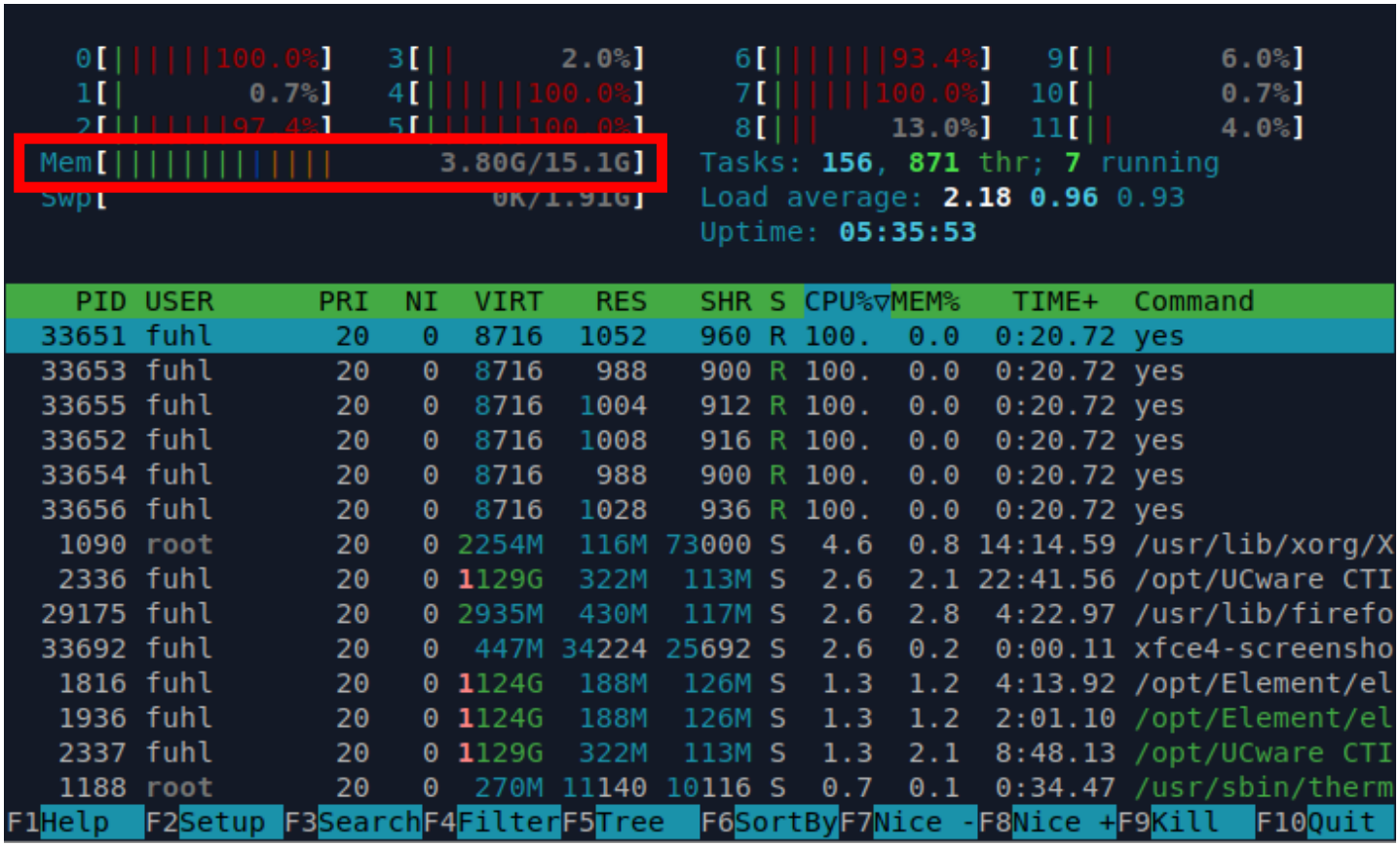
Load average for past one, five, and fifteen minutes

htop — An Alternative to top



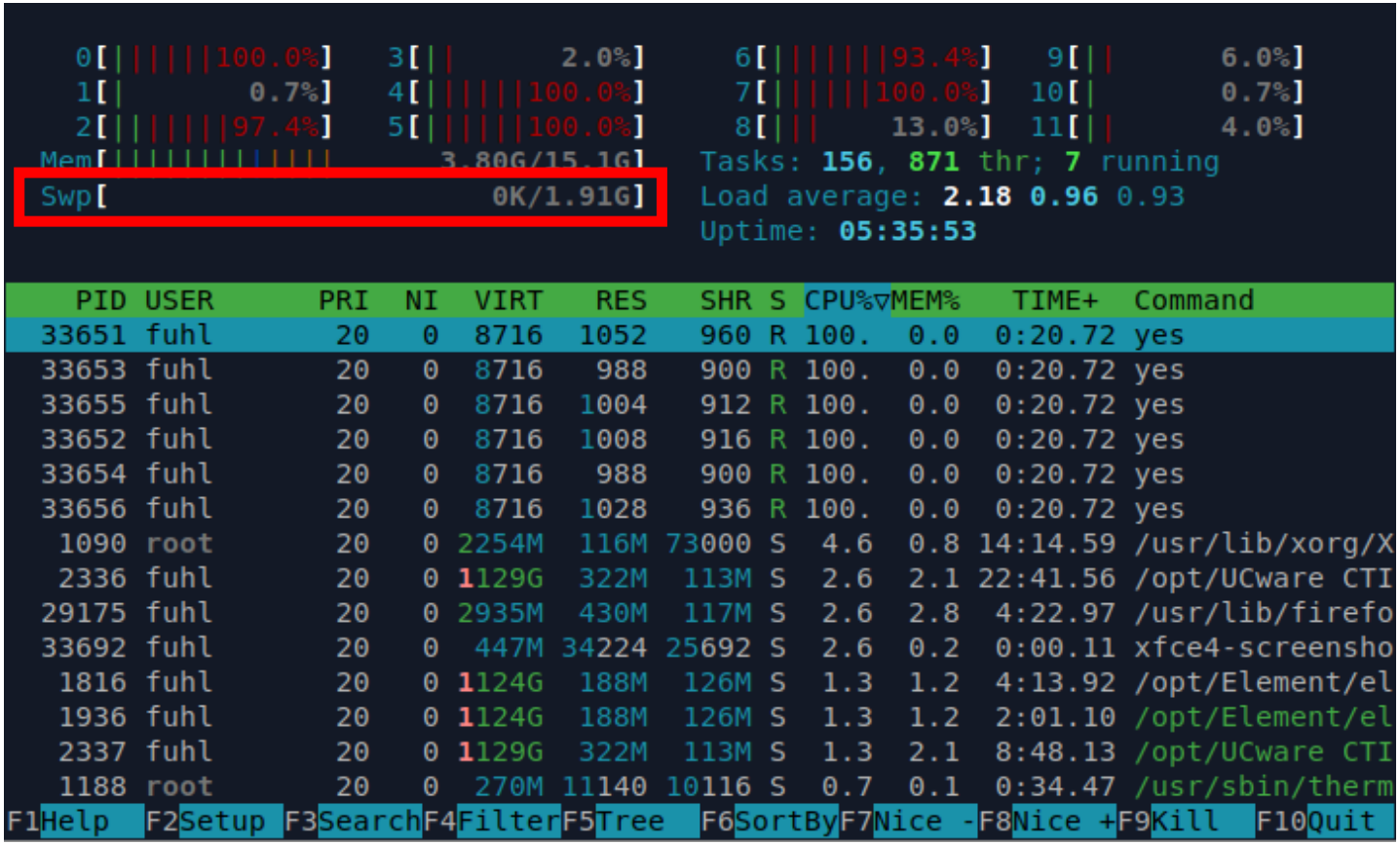
Load on each CPU-core

htop — An Alternative to top



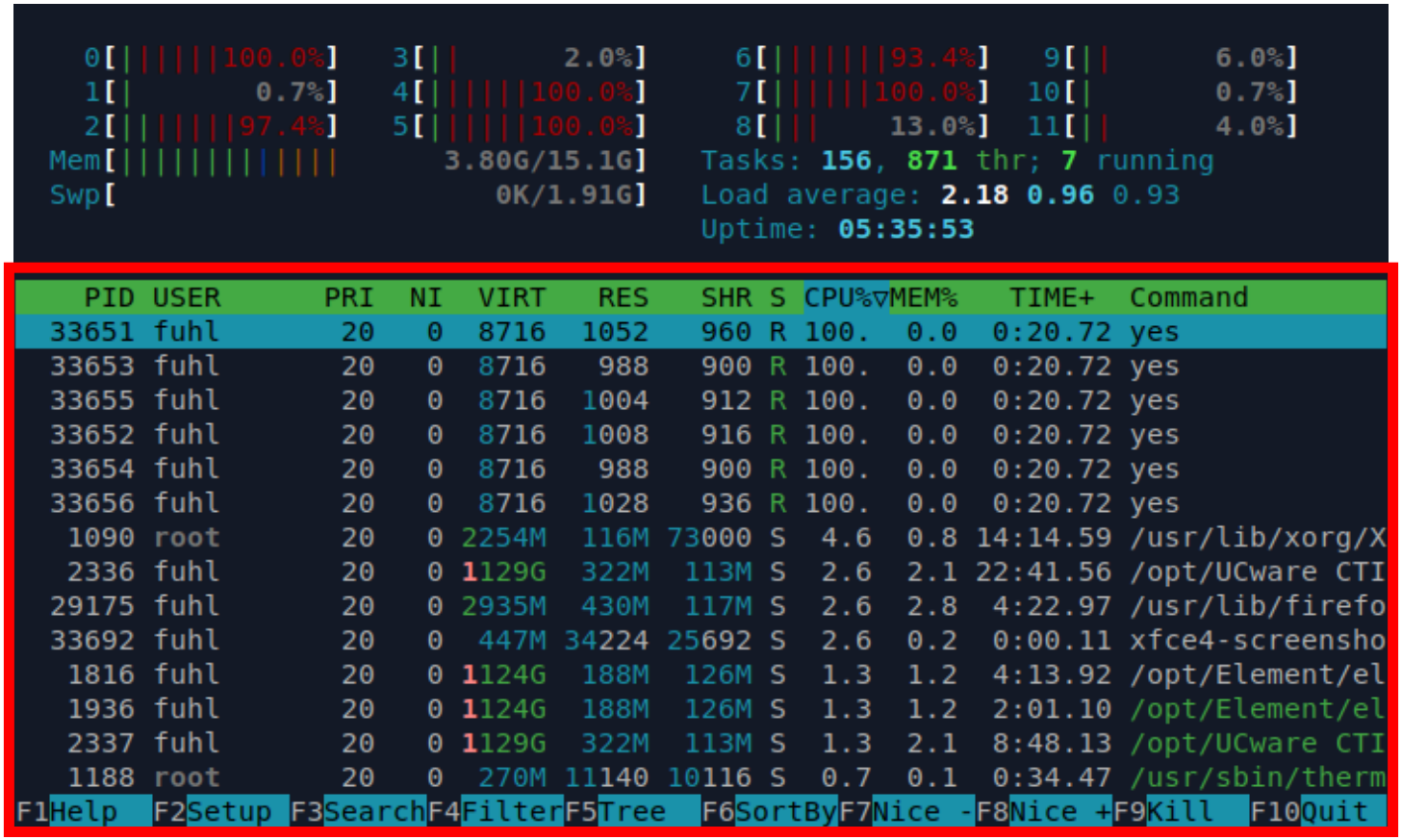
Memory load

htop — An Alternative to top



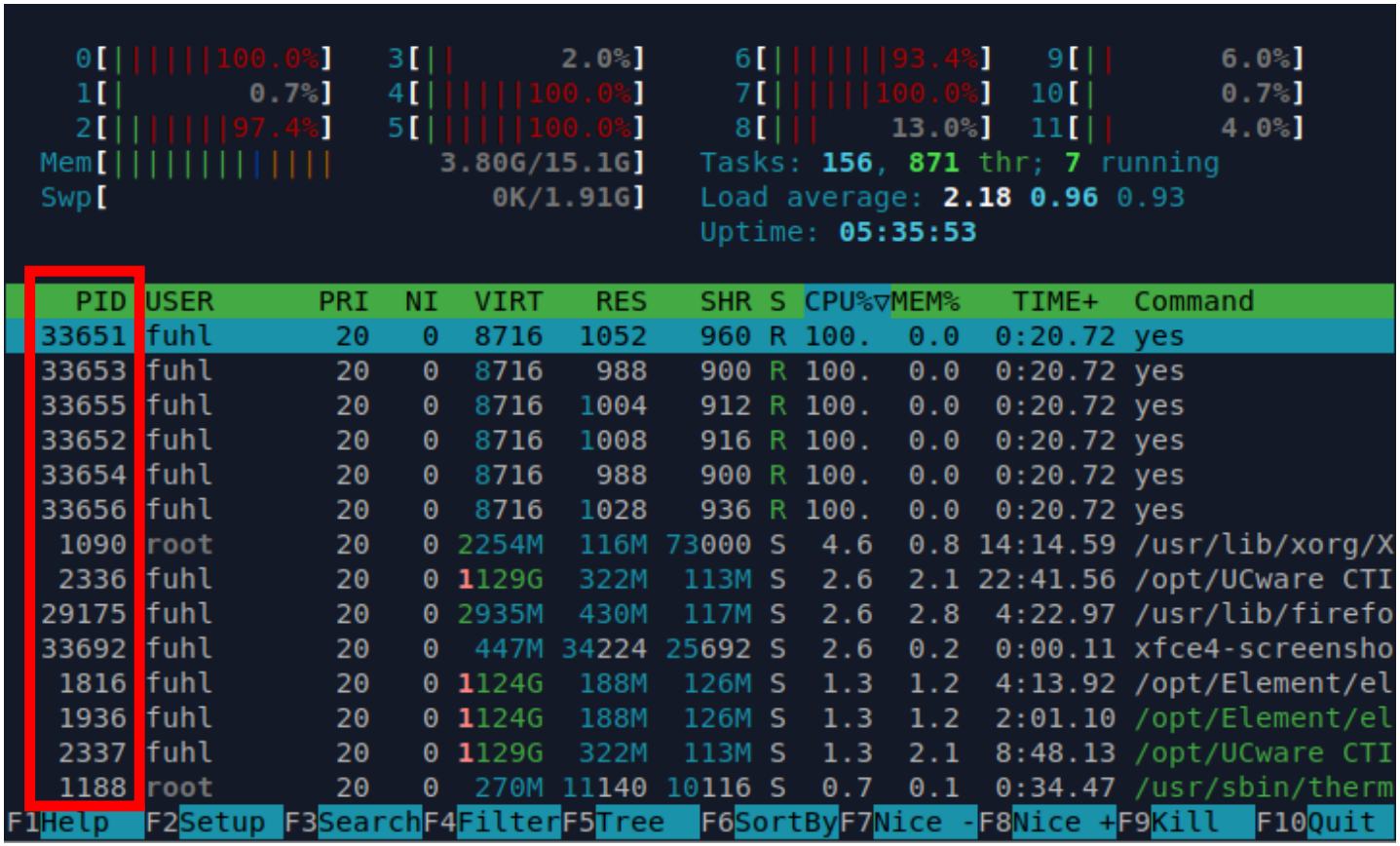
Swap load (try to avoid any swap load)

htop — An Alternative to top



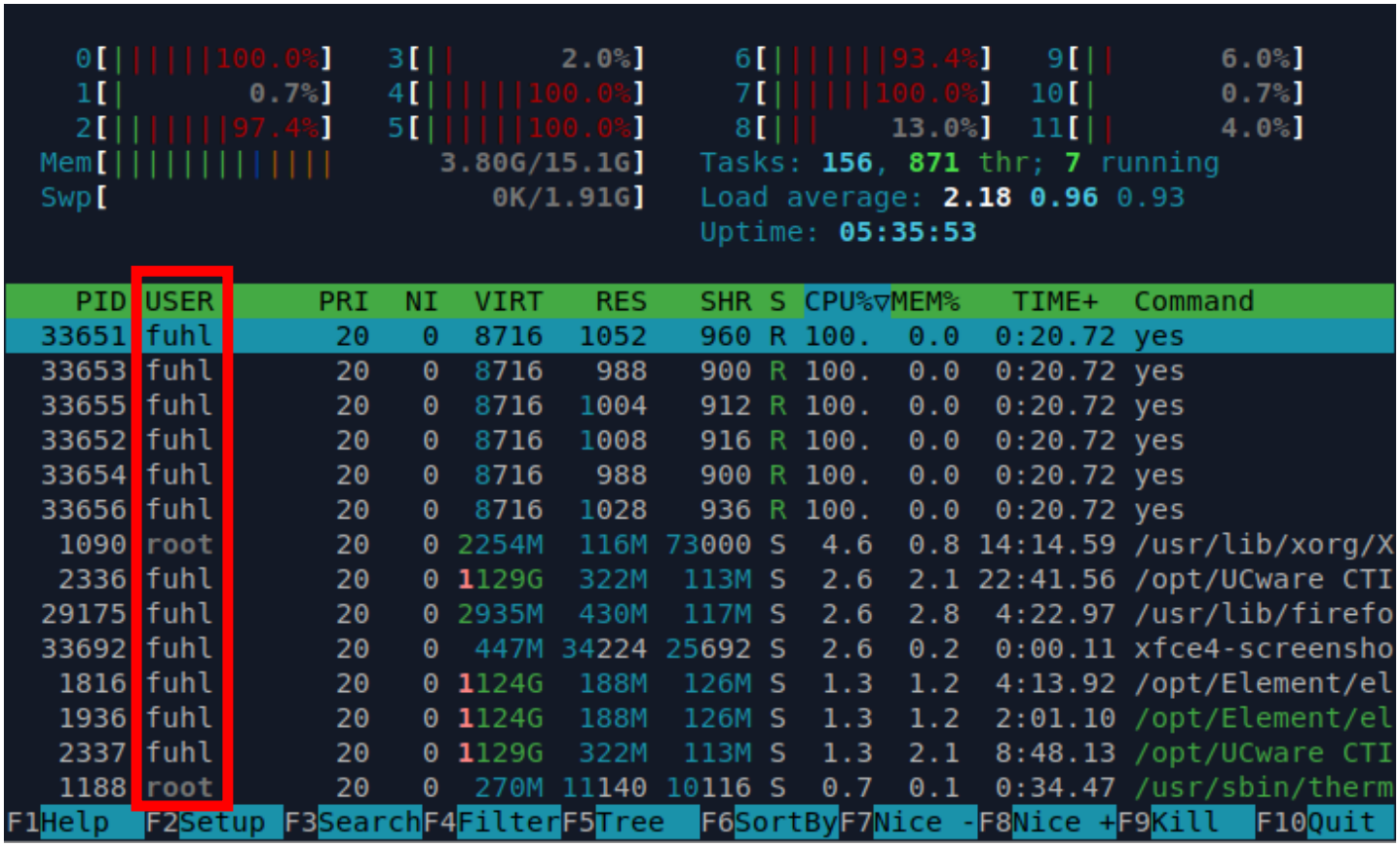
List of runnig processec

htop — An Alternative to top



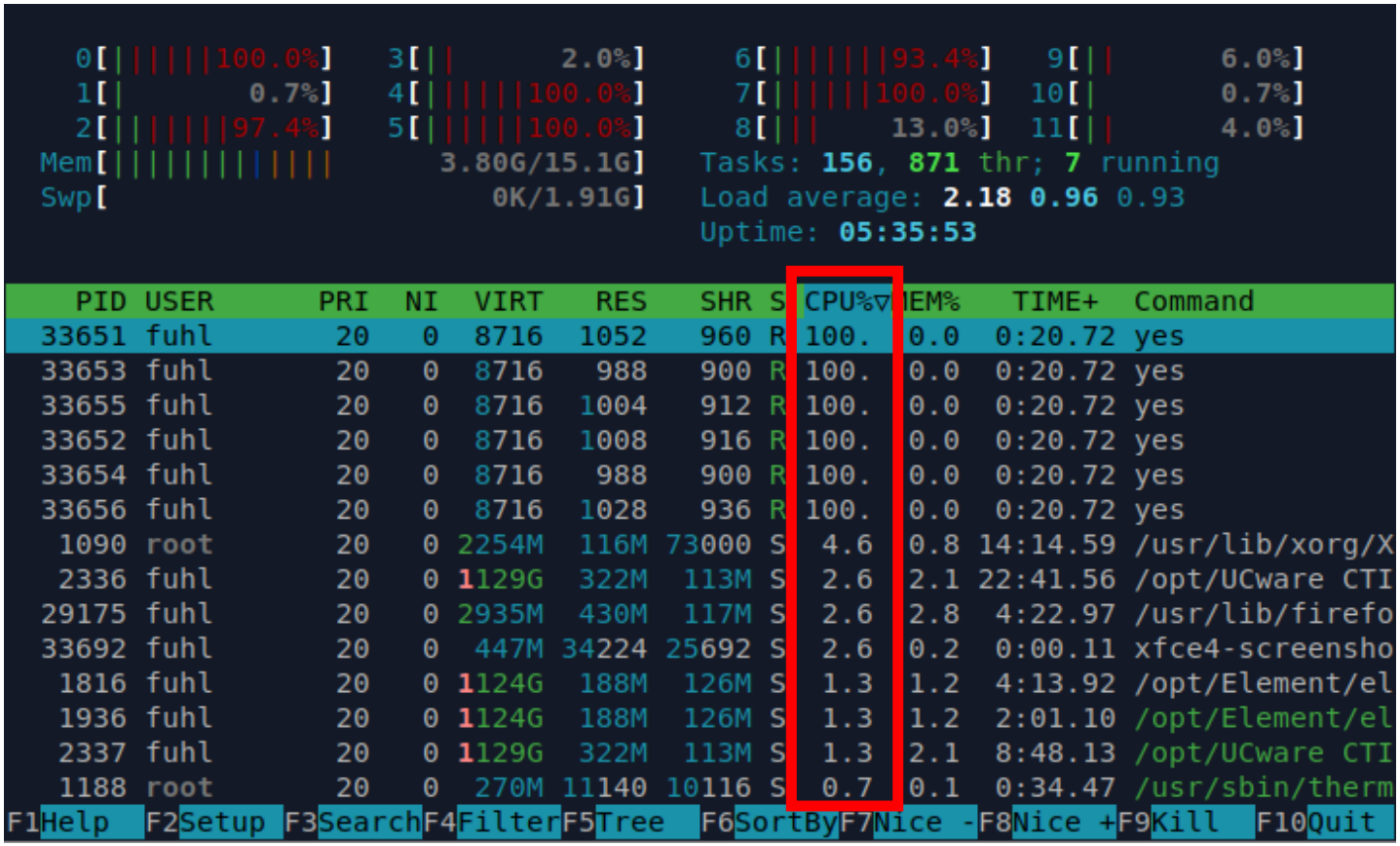
Process ID

htop — An Alternative to top



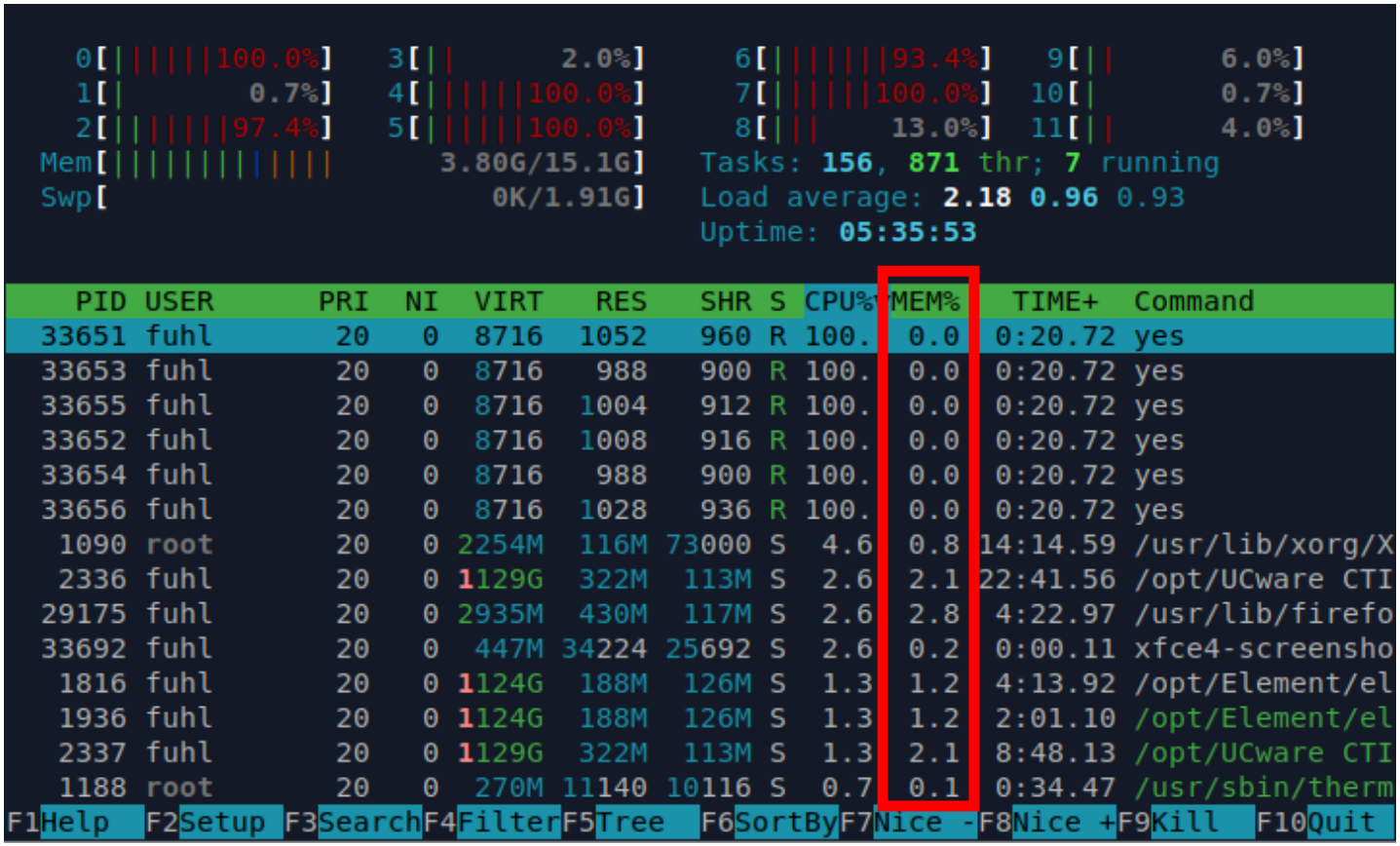
Process owner

htop — An Alternative to top



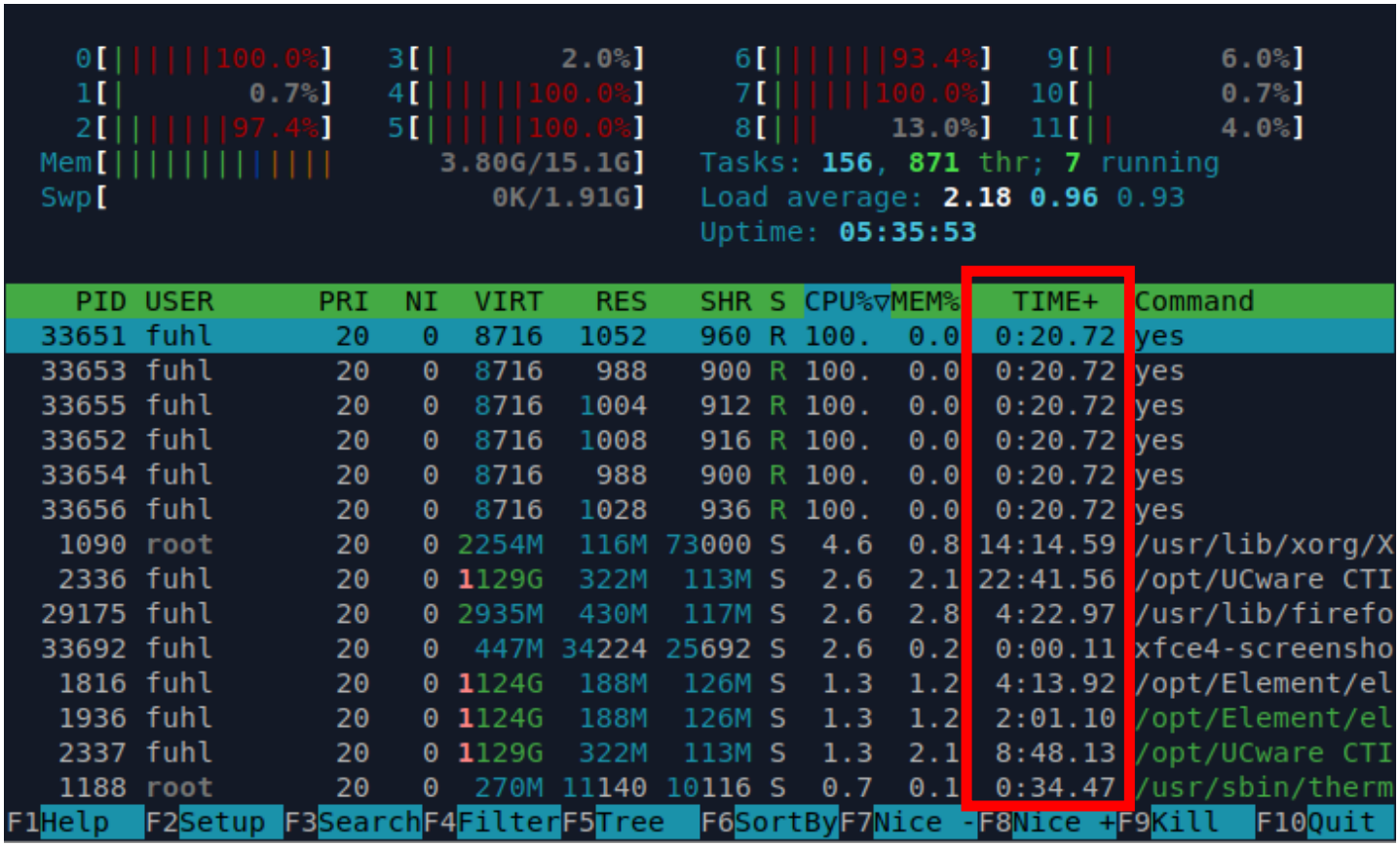
CPU load from this process

htop — An Alternative to top



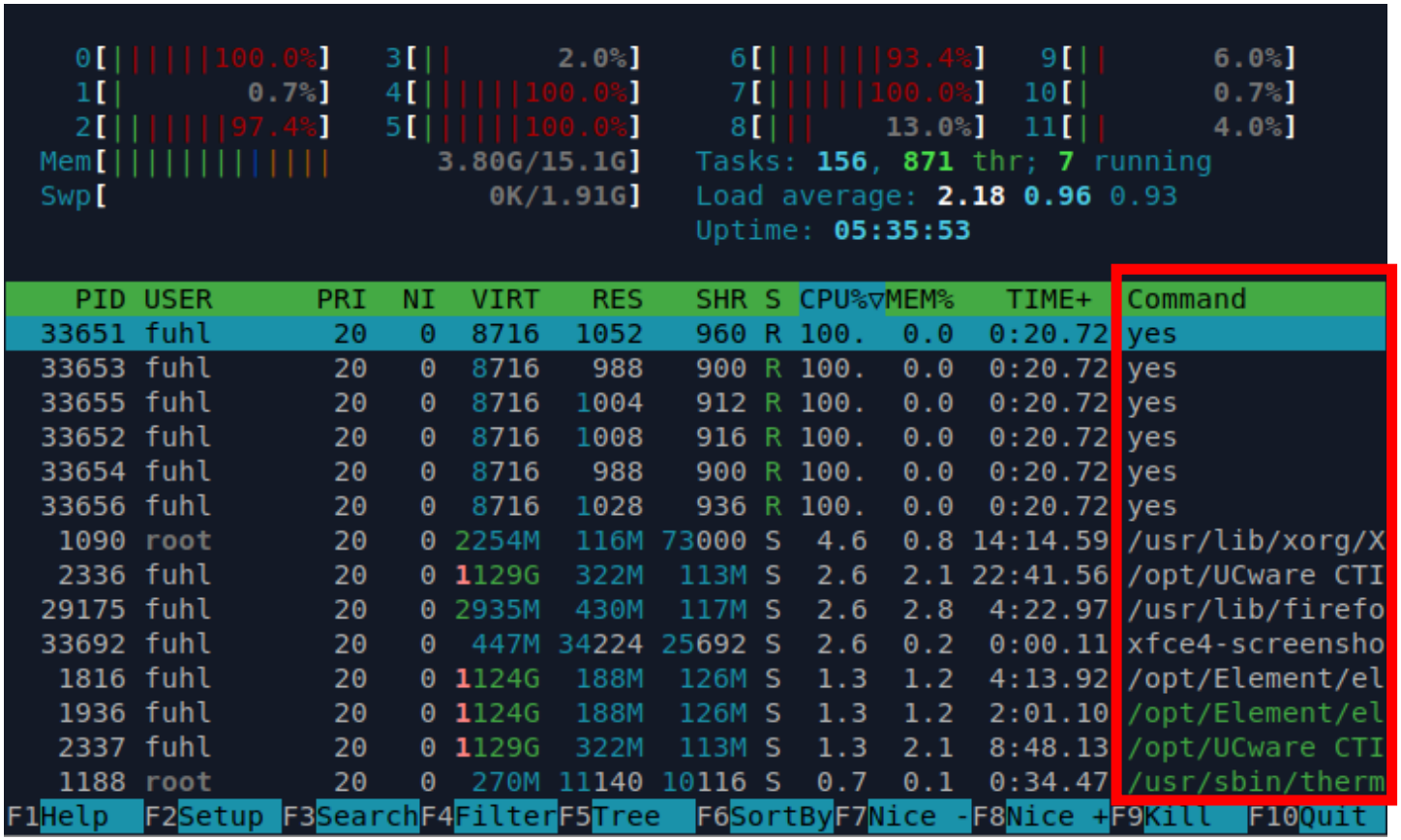
Memory usage of this process

htop — An Alternative to top



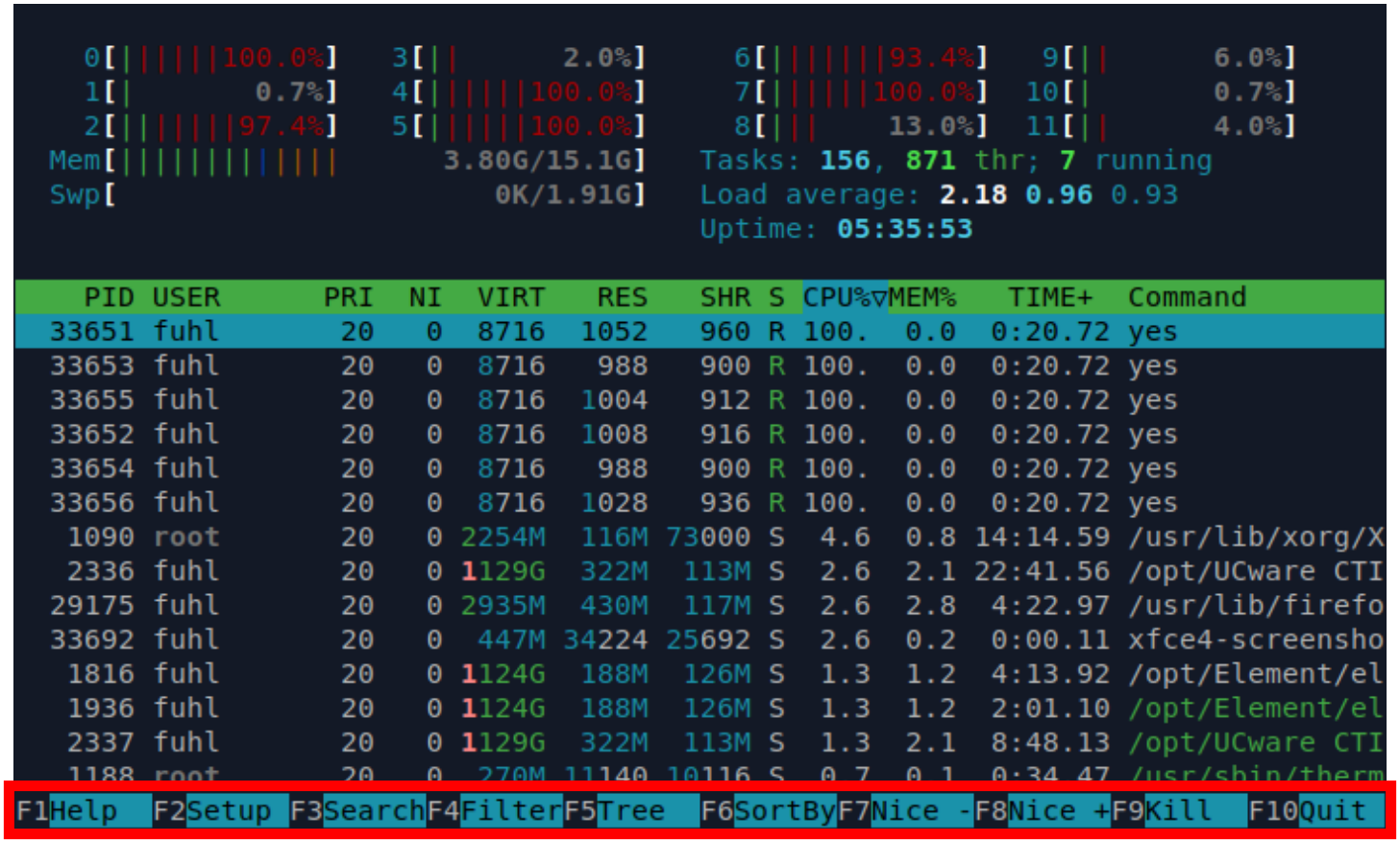
Time since process started

htop — An Alternative to top



Command that started the process

htop — An Alternative to top



Hotkeys

Take Home Messages

Take Home Messages

RUB

- Linux does what you say, not what you mean!

Take Home Messages

RUB

- Linux does what you say, not what you mean!
- Linux does not ask for confirmation!

Take Home Messages

RUB

- Linux does what you say, not what you mean!
- Linux does not ask for confirmation!
- Linux assumes you know what you are doing!

Take Home Messages

RUB

- Linux does what you say, not what you mean!
- Linux does not ask for confirmation!
- Linux assumes you know what you are doing!
- The terminal is a powerful tool! Learn to use it!

Take Home Messages

RUB

- Linux does what you say, not what you mean!
- Linux does not ask for confirmation!
- Linux assumes you know what you are doing!
- The terminal is a powerful tool! Learn to use it!
- Be lazy and learn how to script!

RUB

Happy Computing!

Further Reading

RUB

Linux Tutorial:
`https://hpc-wiki.info/hpc/Introduction_to_Linux_in_HPC`
Linux Cheat Sheet:
`https://linuxconfig.org/linux-commands-cheat-sheet`