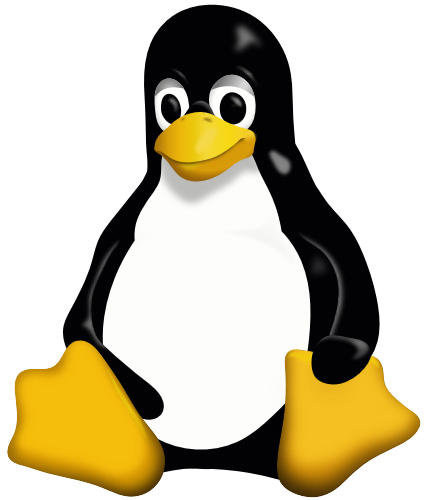
Leiden University
Medical Center

Practical Linux

Core 1

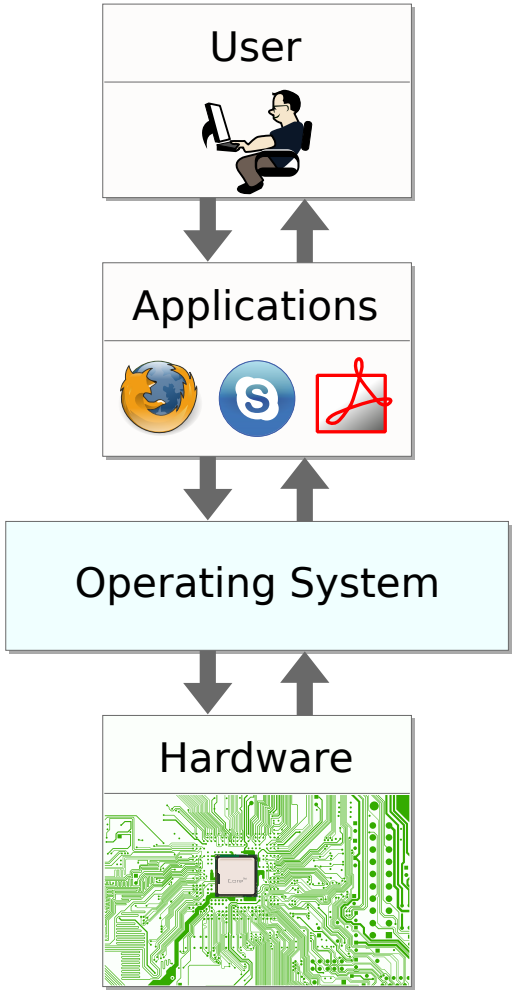


Department of Human Genetics

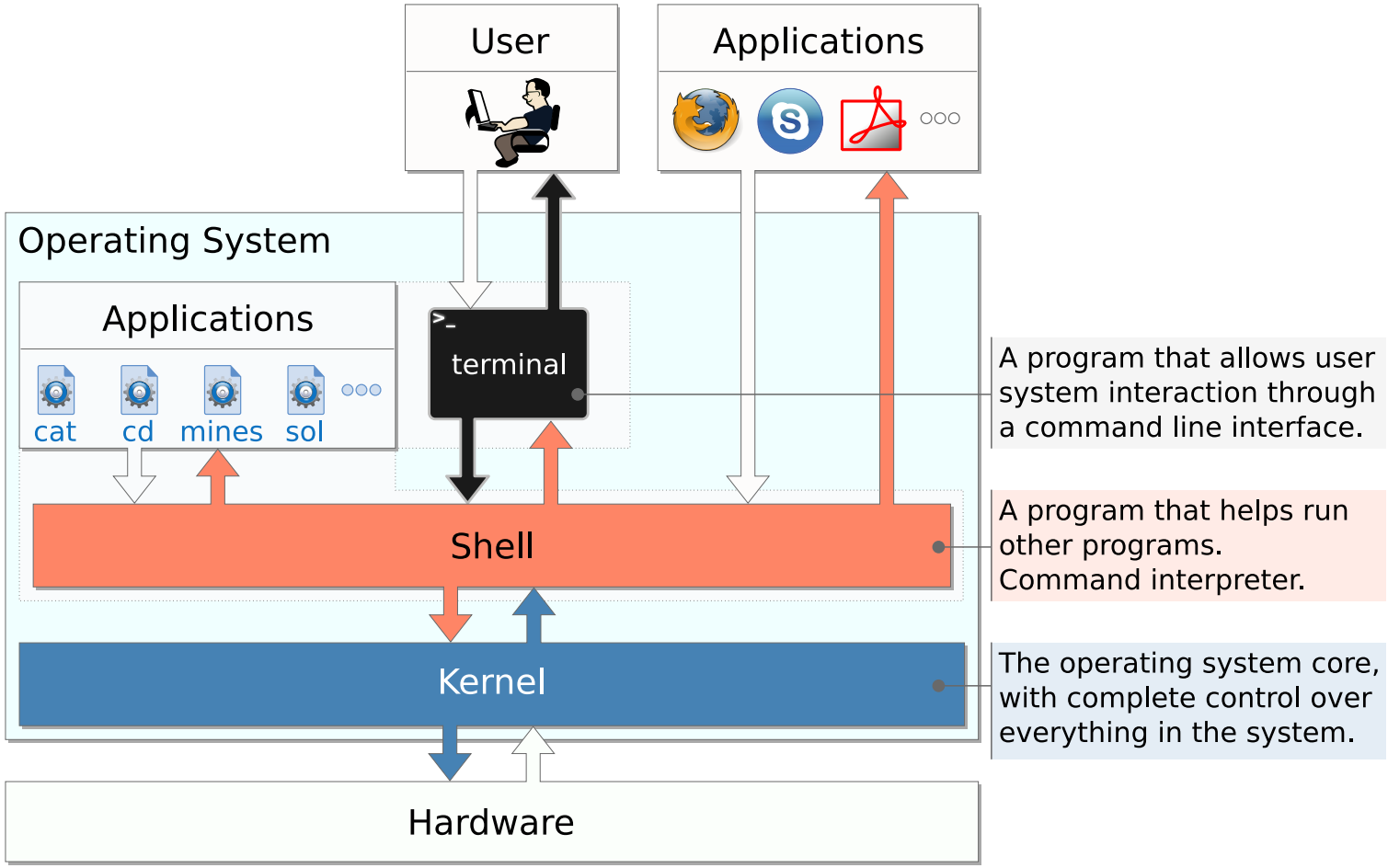
Outline

- The filesystem
- The shell
- The command line interface
- Navigate through the filesystem

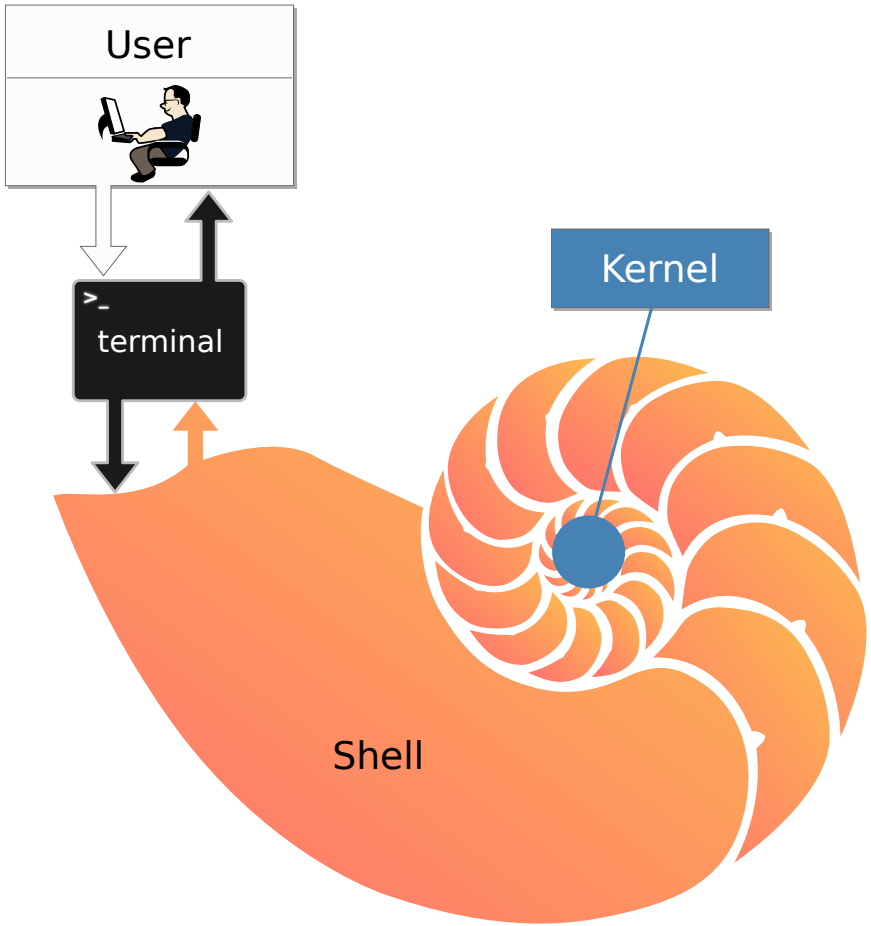
Introduction



The big picture



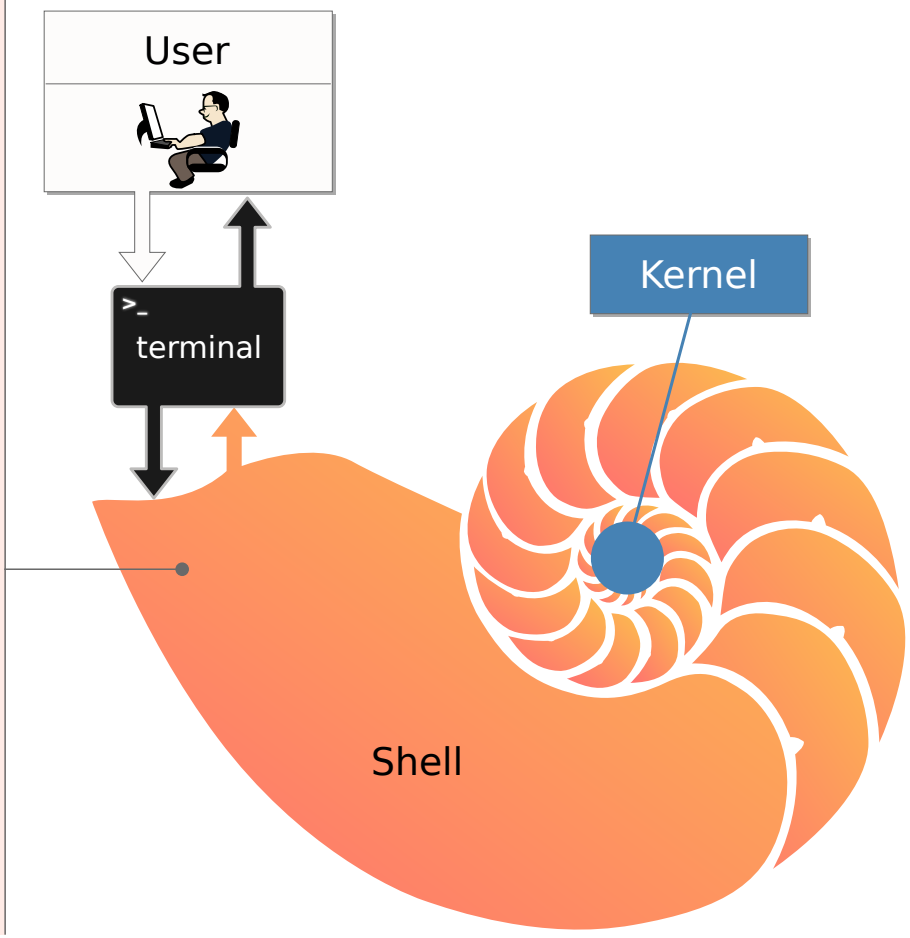
The big picture



The big picture

An ordinary program, you can choose the one that you like. Examples:

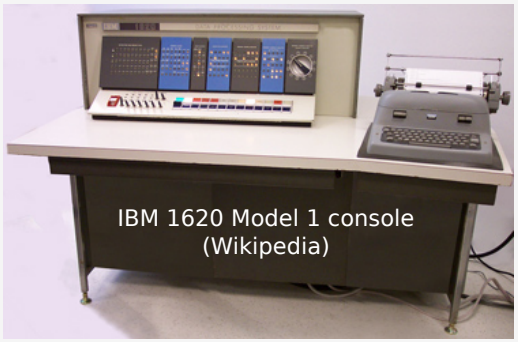
- **sh** The basic shell: the original small program shell with few features, still used on UNIX systems.
- **bash** The Bourne-Again shell: can be found installed and is the default interactive shell on most Linux systems.
- **csh** Its syntax resembles that of the C programming language.
- **tcsh** TENEX C shell: a superset of the common C shell, enhancing user-friendliness and speed (Turbo C shell).
- **ksh** The Korn shell: a superset of the Bourne shell (a nightmare for beginners).
- **zsh** The Z shell: an extended Bourne shell including some features of ksh, and tcsh.
- **...**

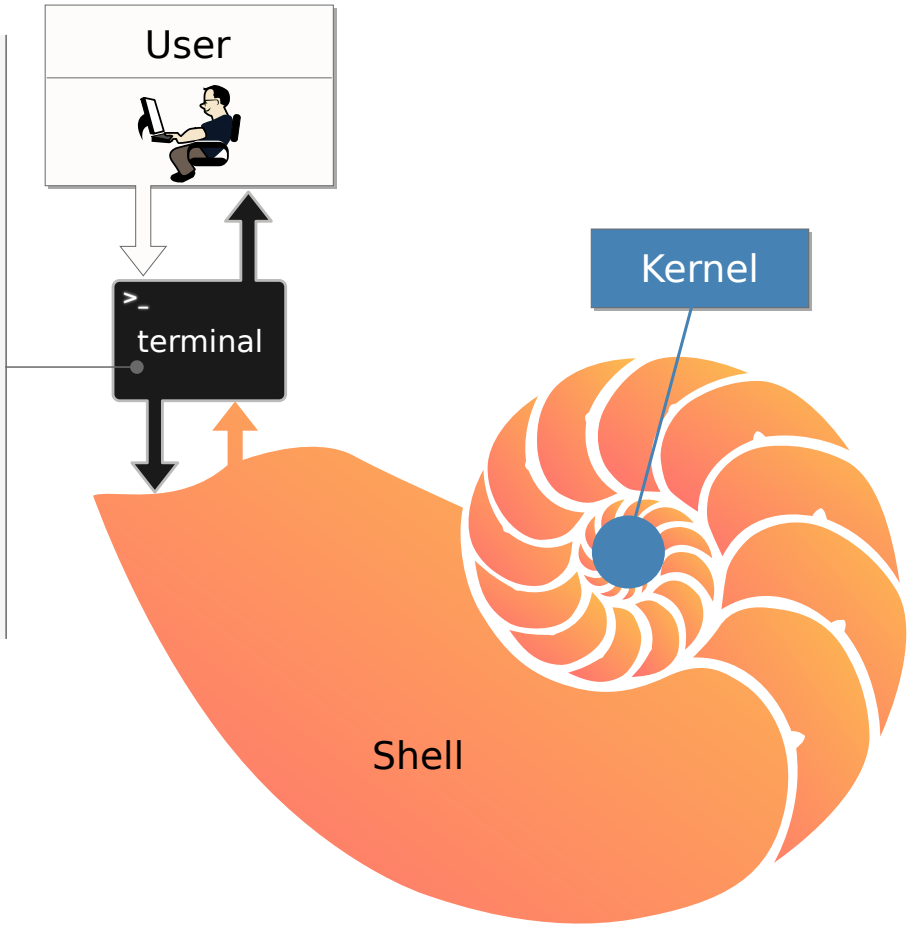


The big picture

History:

- a piece of equipment through which you could interact with a computer.
- teletypewriter - "tty".
- console - furniture point of view:


IBM 1620 Model 1 console
(Wikipedia)

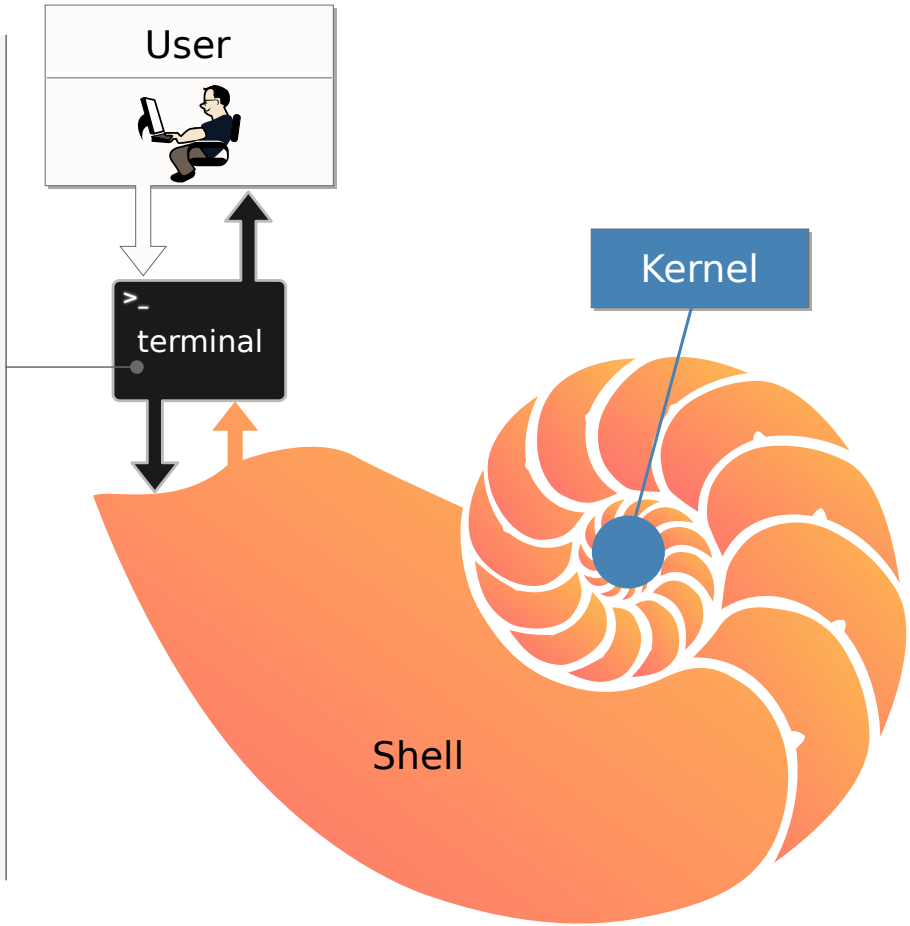


The big picture

History:

- a piece of equipment through which you could interact with a computer.
- teletypewriter - "tty".
- console - furniture point of view.
- terminal - electronic point of view:

DEC VT100 terminal
(Wikipedia)



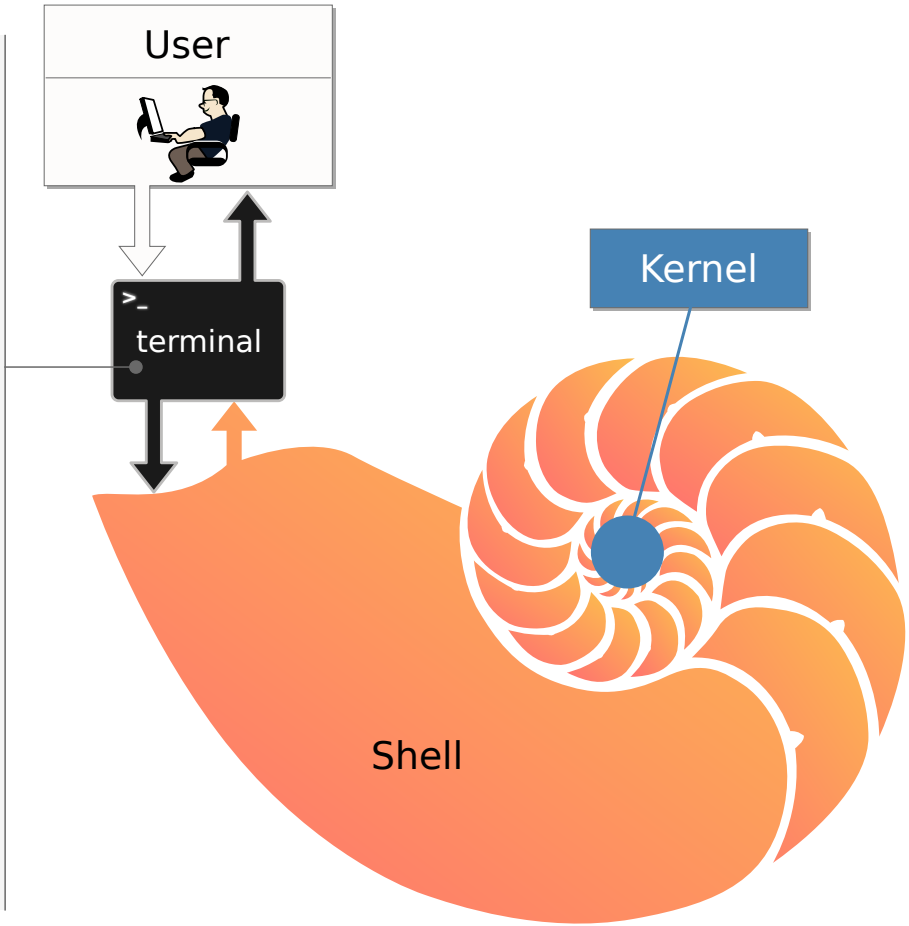
The big picture

History:

- a piece of equipment through which you could interact with a computer.
- teletypewriter - "tty".
- console - furniture point of view.
- terminal - electronic point of view.

Today:

- GUI terminal emulators:
  - Terminal, GNOME Terminal, XTerm, KDE Konsole, ...



Command prompt

Command prompt:
- a short text message at the start of the command line.

The default prompt on the bash shell contains:
- user name
- computer name
- current directory path

Command prompt

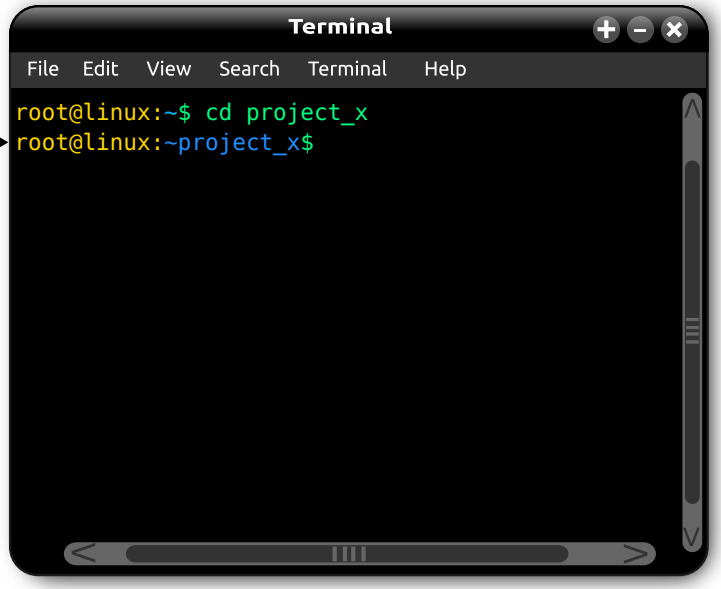
Command prompt:
- a short text message at the start of the command line.

The default prompt on the bash shell contains:
- user name
- computer name
- current directory path

Command prompt

Command prompt:
- a short text message at the start of the command line.

The default prompt on the bash shell contains:
- user name
- computer name
- current directory path



Command prompt

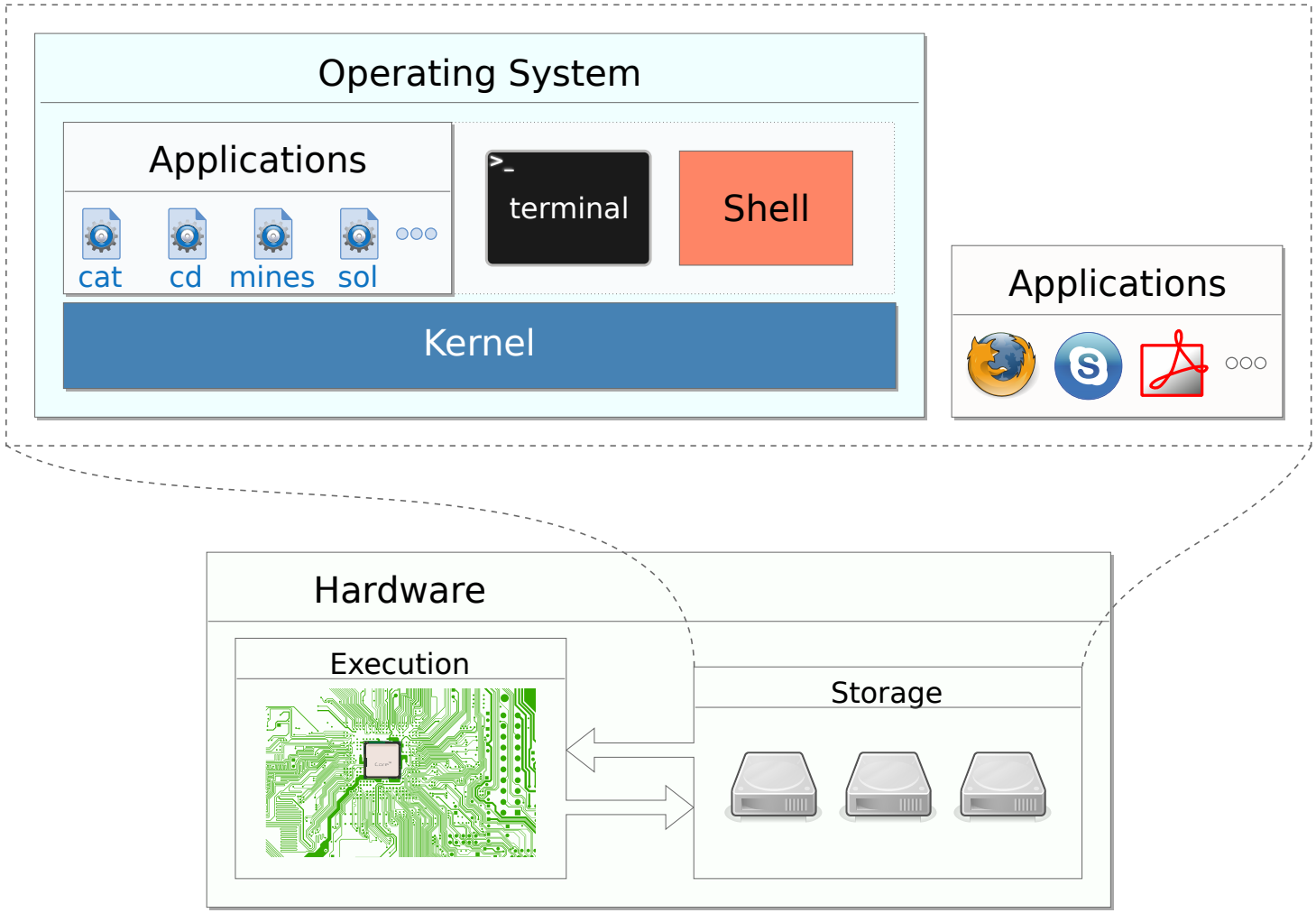
We use here an empty prompt:

```
$ cd project_x
$
```

Maneuvering

- Command completion help: `TAB` key.
- Stop entering line: `Ctrl-c`, i.e., `Ctrl` and `c` keys.
  - Note that `Ctrl-c` is not used for copy.
- Navigate the command line with $\leftarrow$ and $\rightarrow$ arrow keys.
- Go to the command line start/end: `HOME` / `END` keys.
- Navigate through history commands with $\uparrow$ and $\downarrow$ arrow keys.
- Quit: `Ctrl-d`.

The manner in which storage is organized



Everything is a file

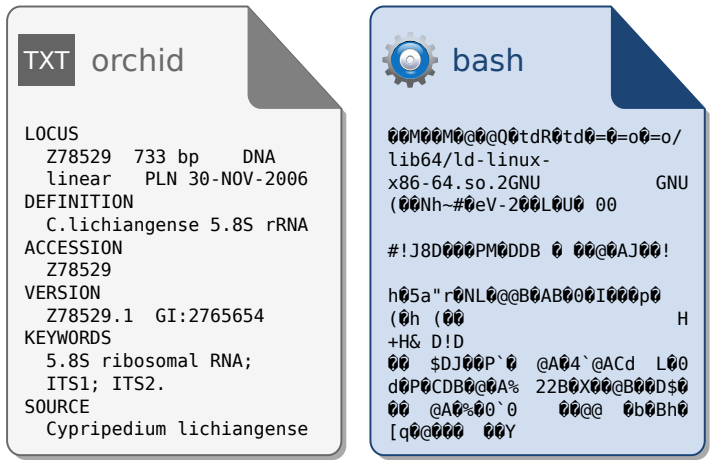
A file:

- A named collection of related data.
- Appears to the user as a single, contiguous information block.
- Is retained in storage.

Everything is a file

A file:

- A named collection of related data.
- Appears to the user as a single, contiguous information block.
- Is retained in storage.



Basic file types:

- **Text files**:
  - Contain human-readable characters and few control characters.
- **Binary files**:
  - Any file that contains at least some binary data:
    - Image files, compressed files, executable programs, etc.
  - Usually entirely not human readable.

File names

- Case sensitive:
  - “bill” is different than “BiLL”.
- No obvious length limit.
- Can contain any characters except for “/”.

File names

- Case sensitive:
  - "bill" is different than "BiLL".
- No obvious length limit.
- Can contain any characters except for "/".
  Important:
  - Whitespaces are allowed, but try to avoid them.
  - Avoid starting a filename with a "." since it will be hidden.
  - File names should **never begin** with a **hyphen ("-")**.
  - Use only **lower case alphanumerical characters** and **underscores**.
    - Other characters, such as $, % [,],(,), have special meanings to the shell and can be distracting to work with.

File names

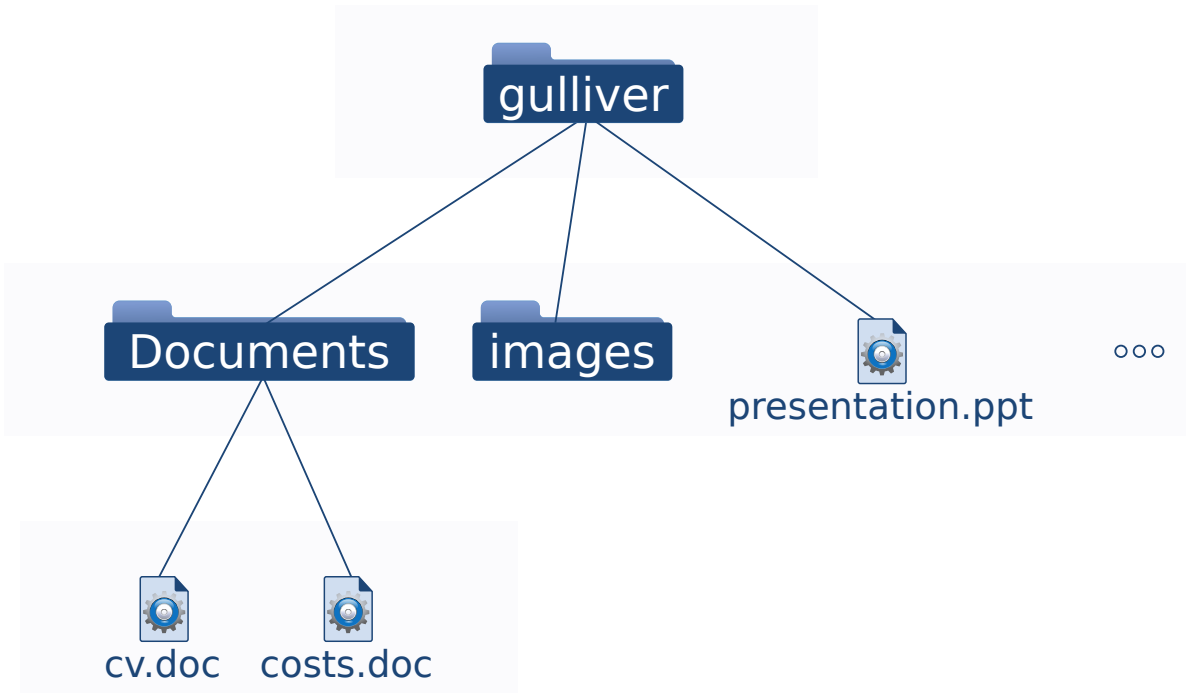
- Case sensitive:
  - "bill" is different than "BiLL".
- No obvious length limit.
- Can contain any characters except for "/".
  Important:
  - Whitespaces are allowed, but try to avoid them.
  - Avoid starting a filename with a "." since it will be hidden.
  - File names should **never begin** with a **hyphen ("-")**.
  - Use only **lower case alphanumerical characters** and **underscores**.
    - Other characters, such as $, % [,],(,), have special meanings to the shell and can be distracting to work with.
- File name extension not needed - used just for convenience.
  - File types stored in the file.

Directory (folder)

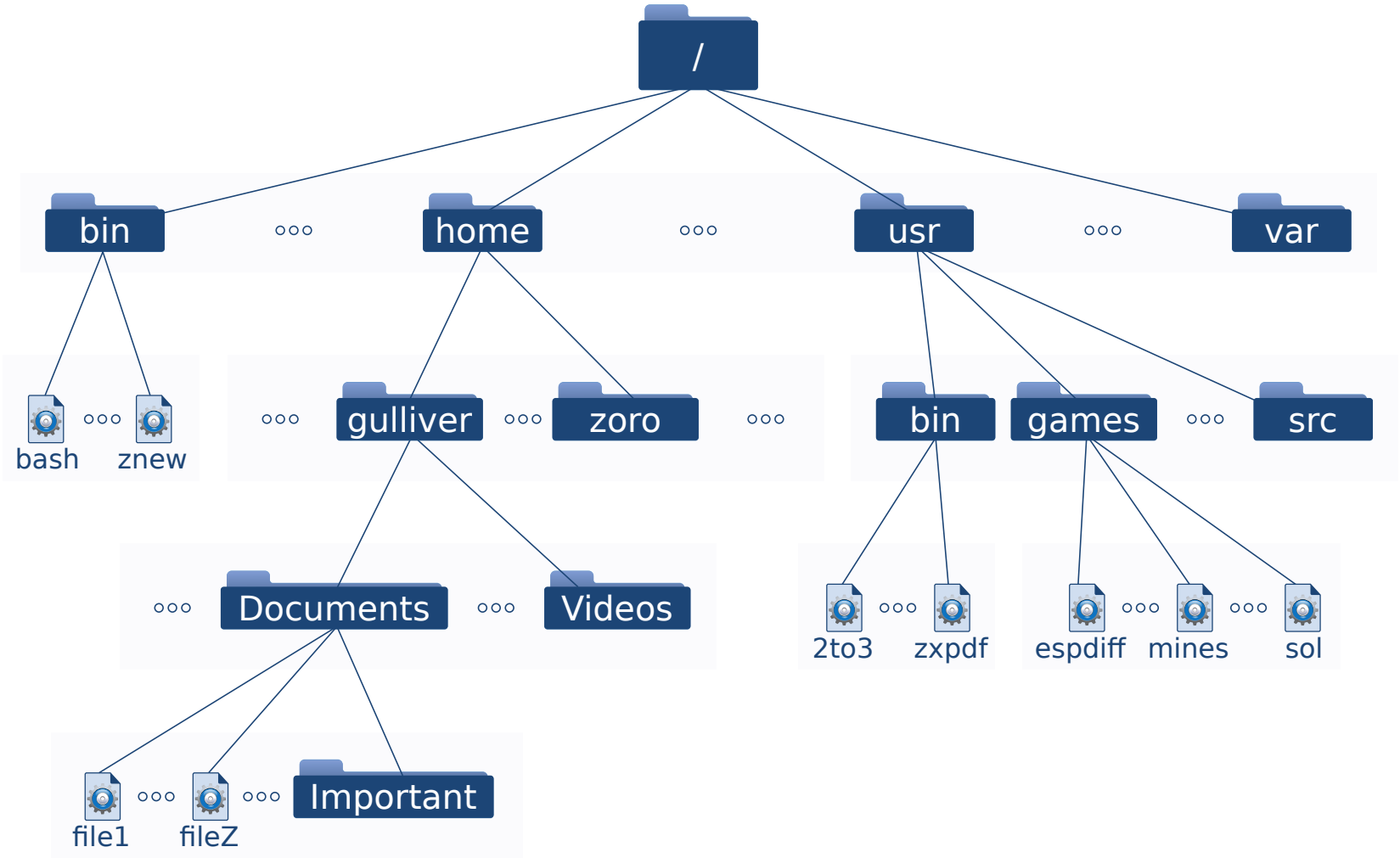
- A **directory** is just a **special file**:
  - A container that can hold files and other directories.

Directory (folder)

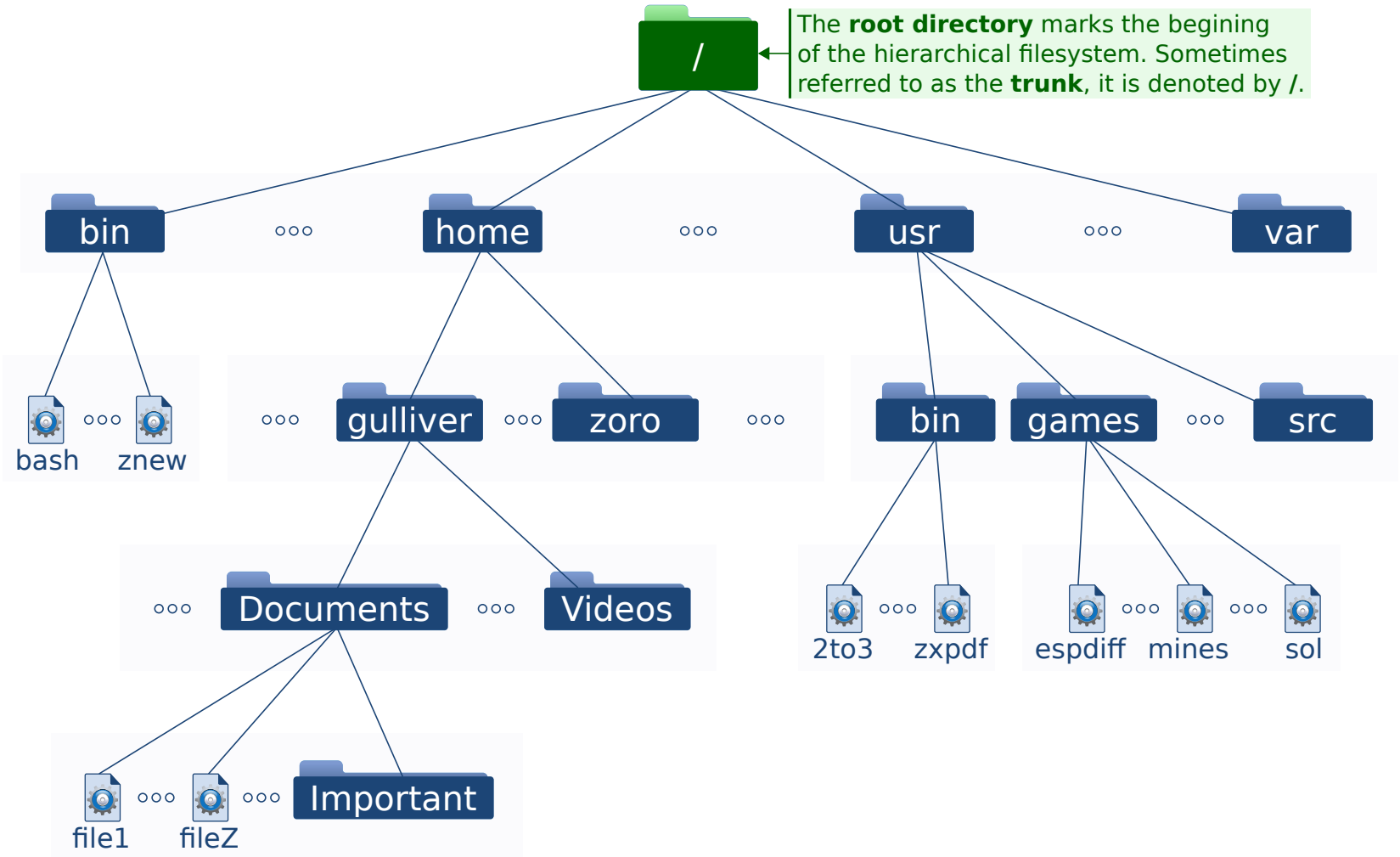
- A **directory** is just a **special file**:
  - A container that can hold files and other directories.



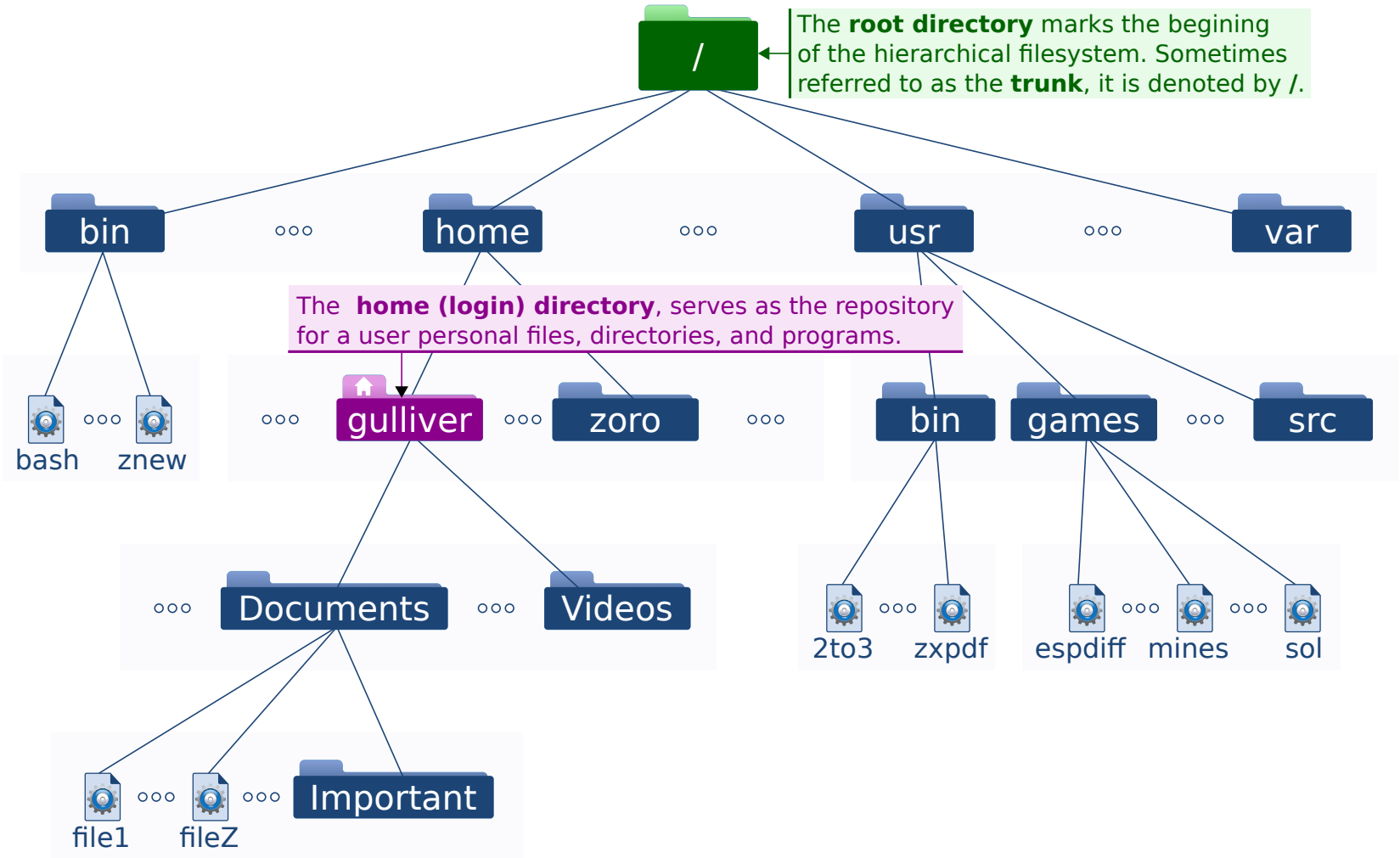
Tree structure



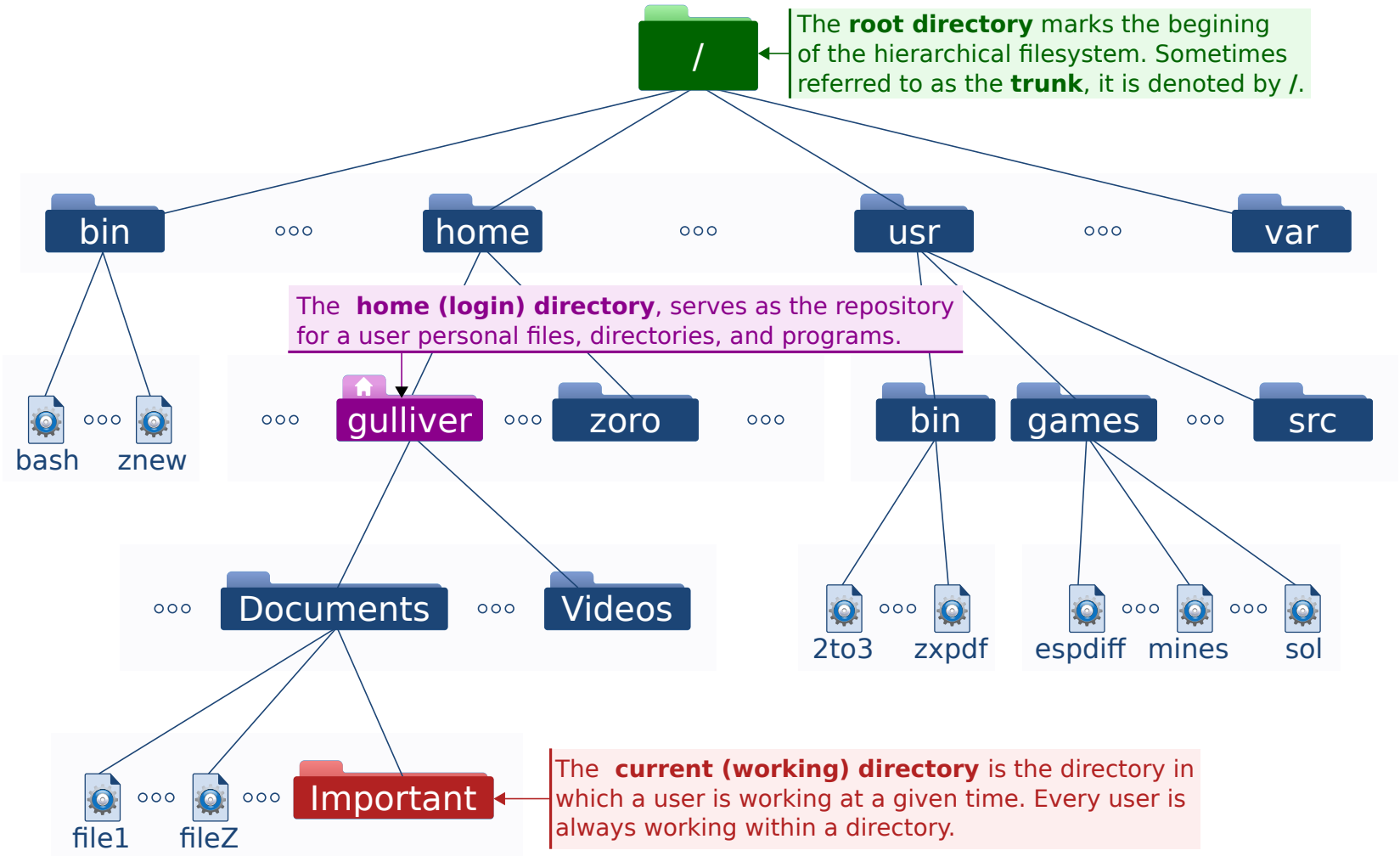
Special directories



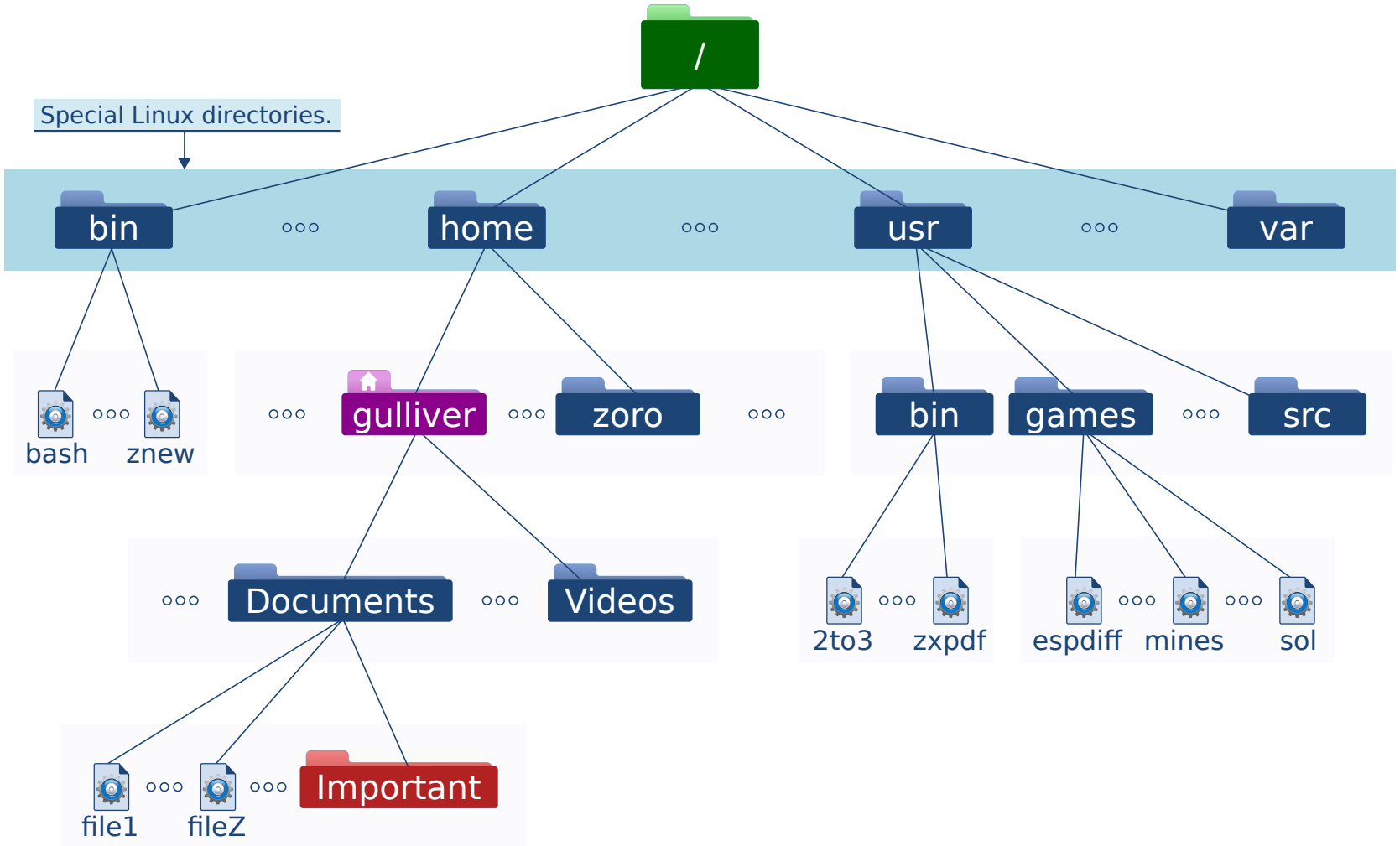
Special directories



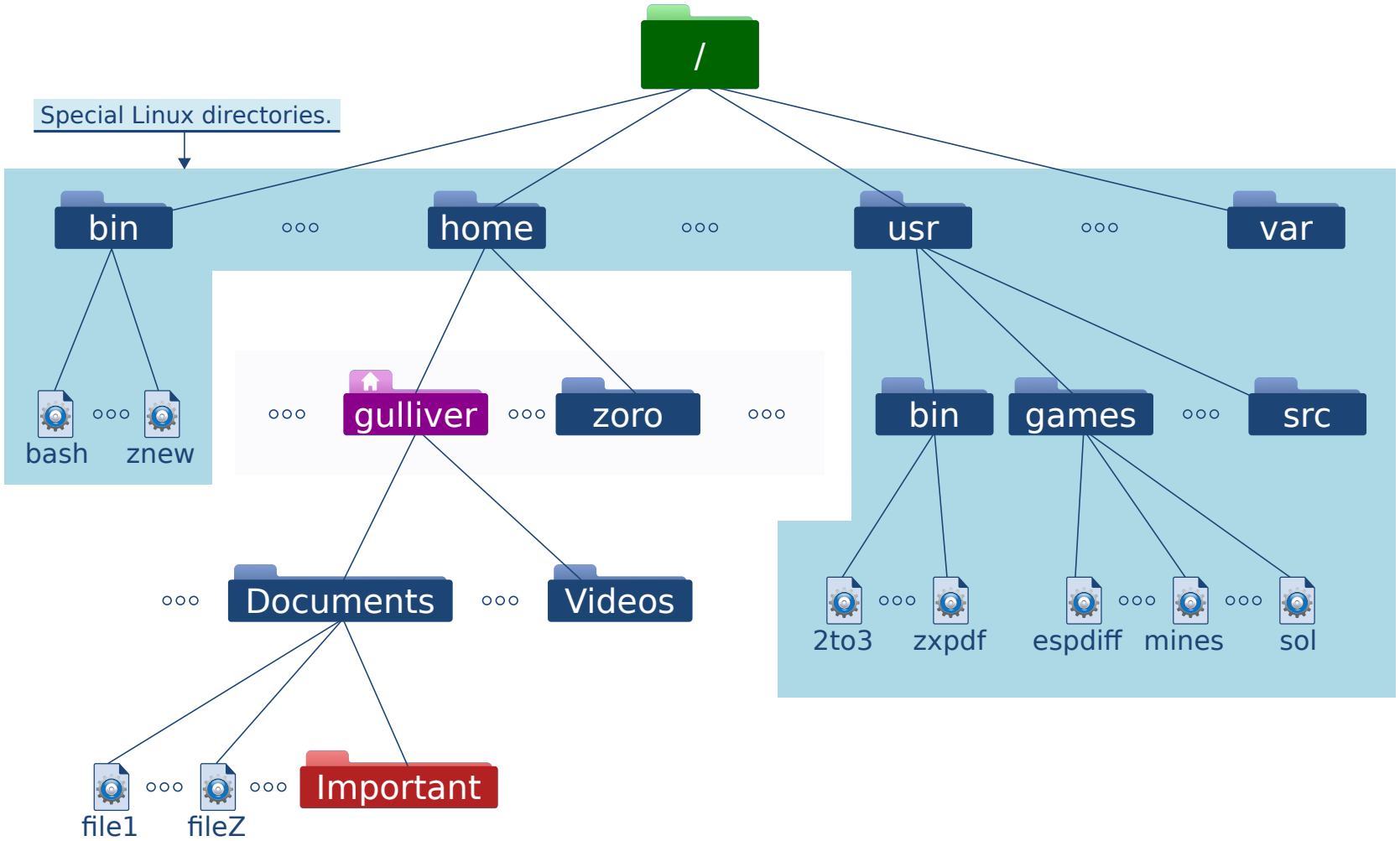
Special directories



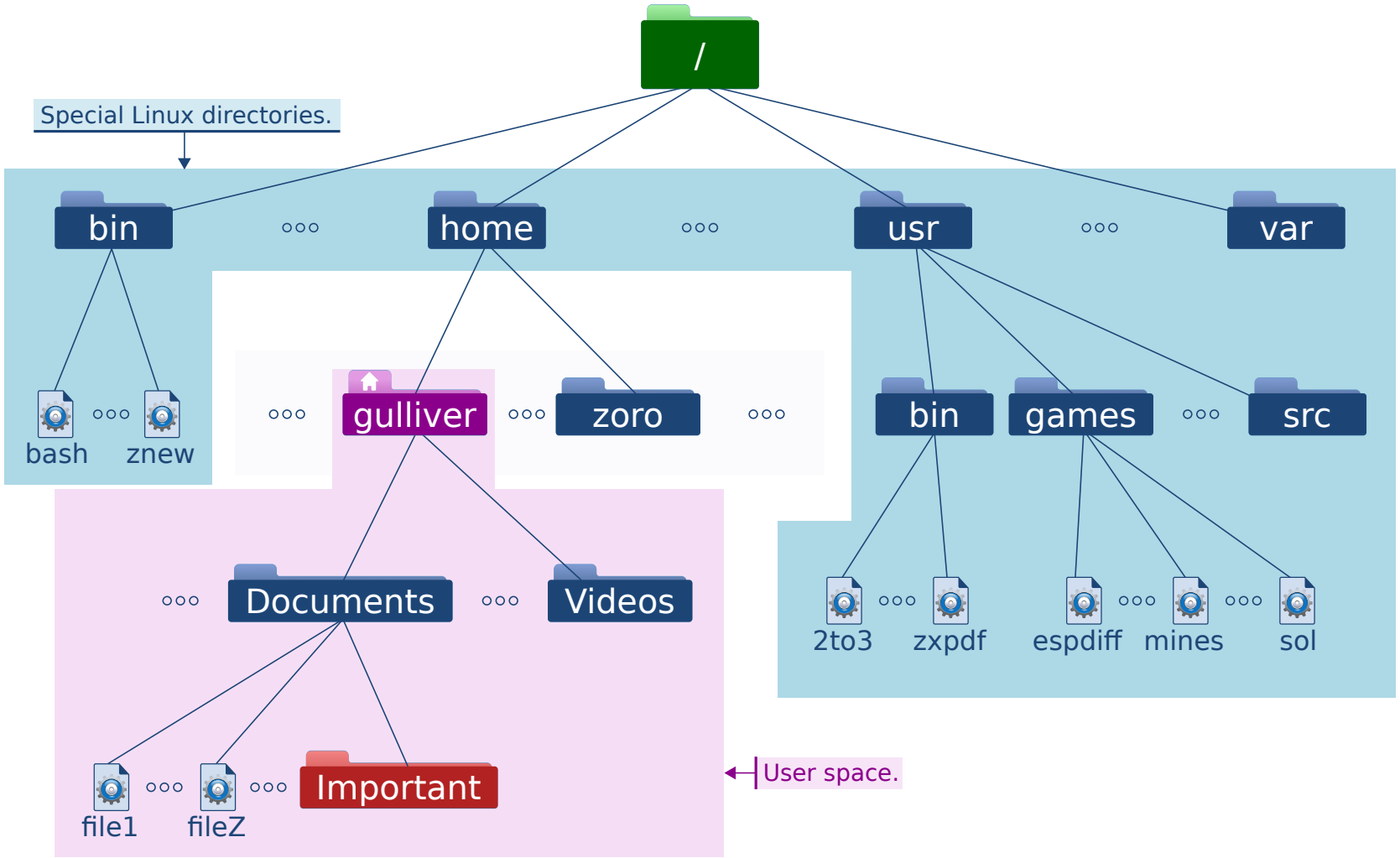
Linux directories



Linux directories



User area


/
Special Linux directories.
bin
home
usr
var
bash
znew
gulliver
zoro
bin
games
src
Documents
Videos
2to3
zxpdf
espdiff
mines
sol
file1
fileZ
Important
User space.

Path

The route along the tree branches to get to the wanted directory.

Components:

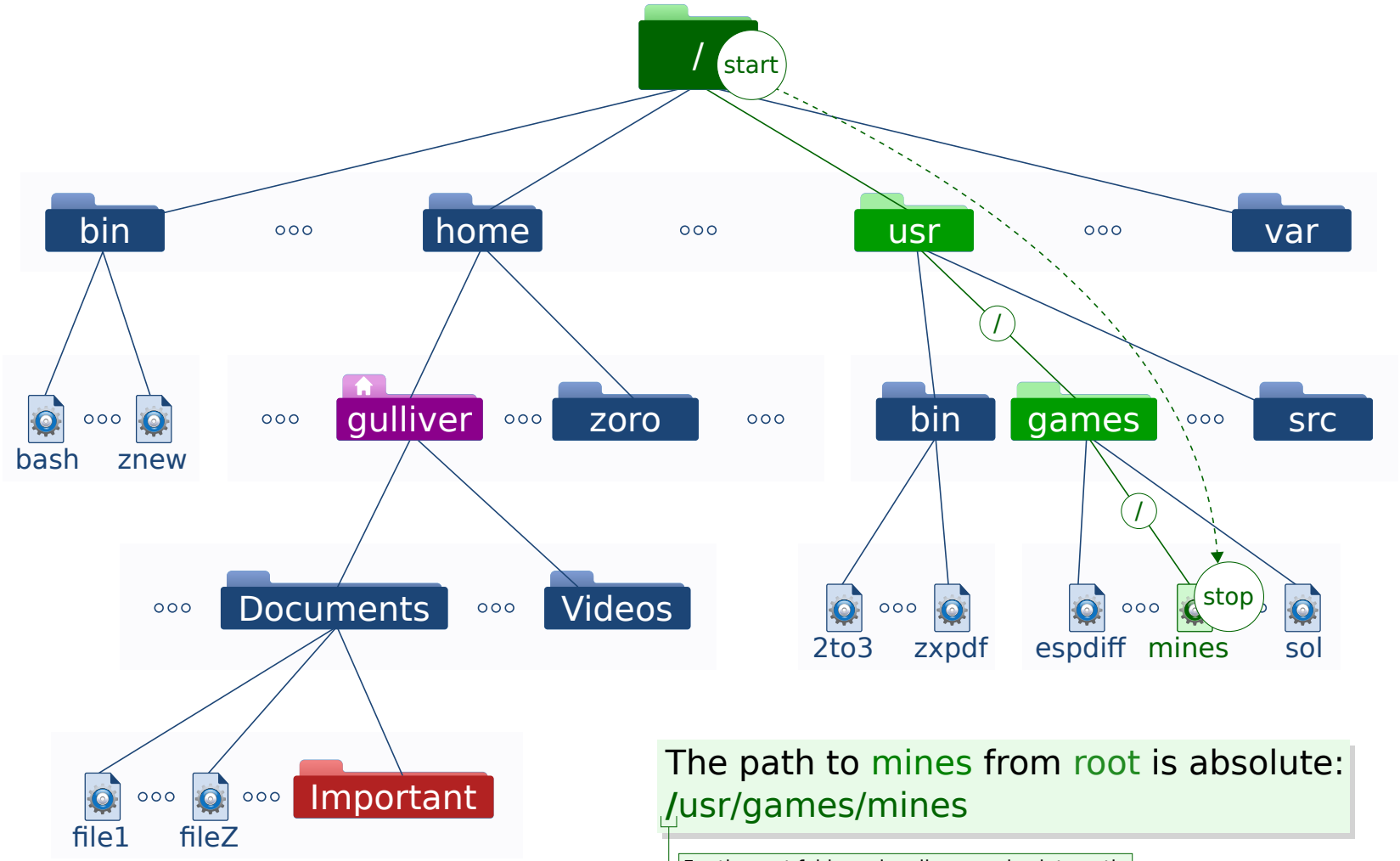
- The directory names encountered.
- The delimiting character: the slash ("/").
- Sometimes one dot (".") to represent the **current directory**.
- Sometimes two dots ("..") to indicate the **parent directory**.
  - Point upwards in the hierarchy.

Examples:

- `/bin/bash`
- `bin/bash`
- `./sol`
- `../../games/mines`

No blank spaces in the path!

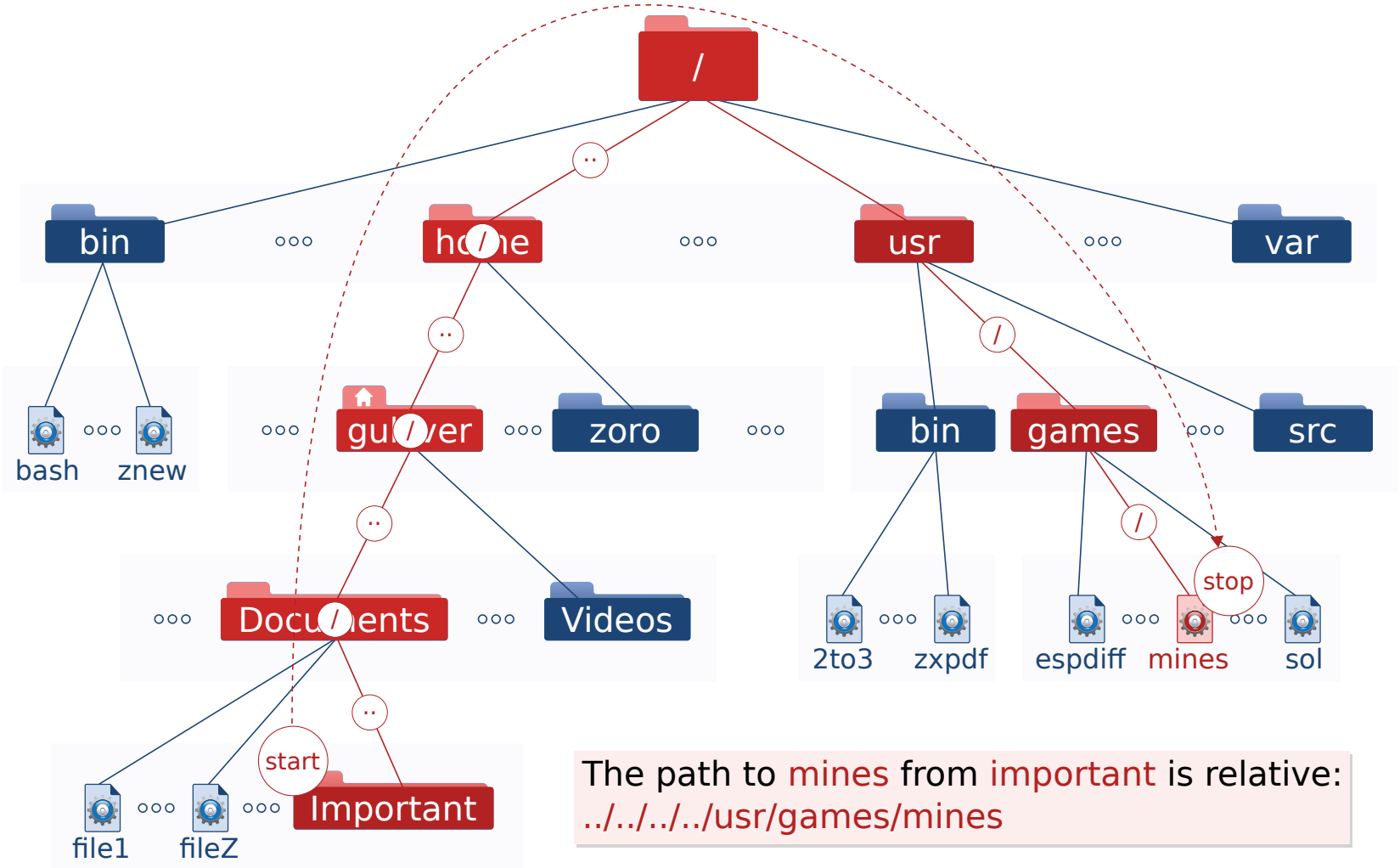
Absolute paths - start from the root



The path to mines from root is absolute: /usr/games/mines

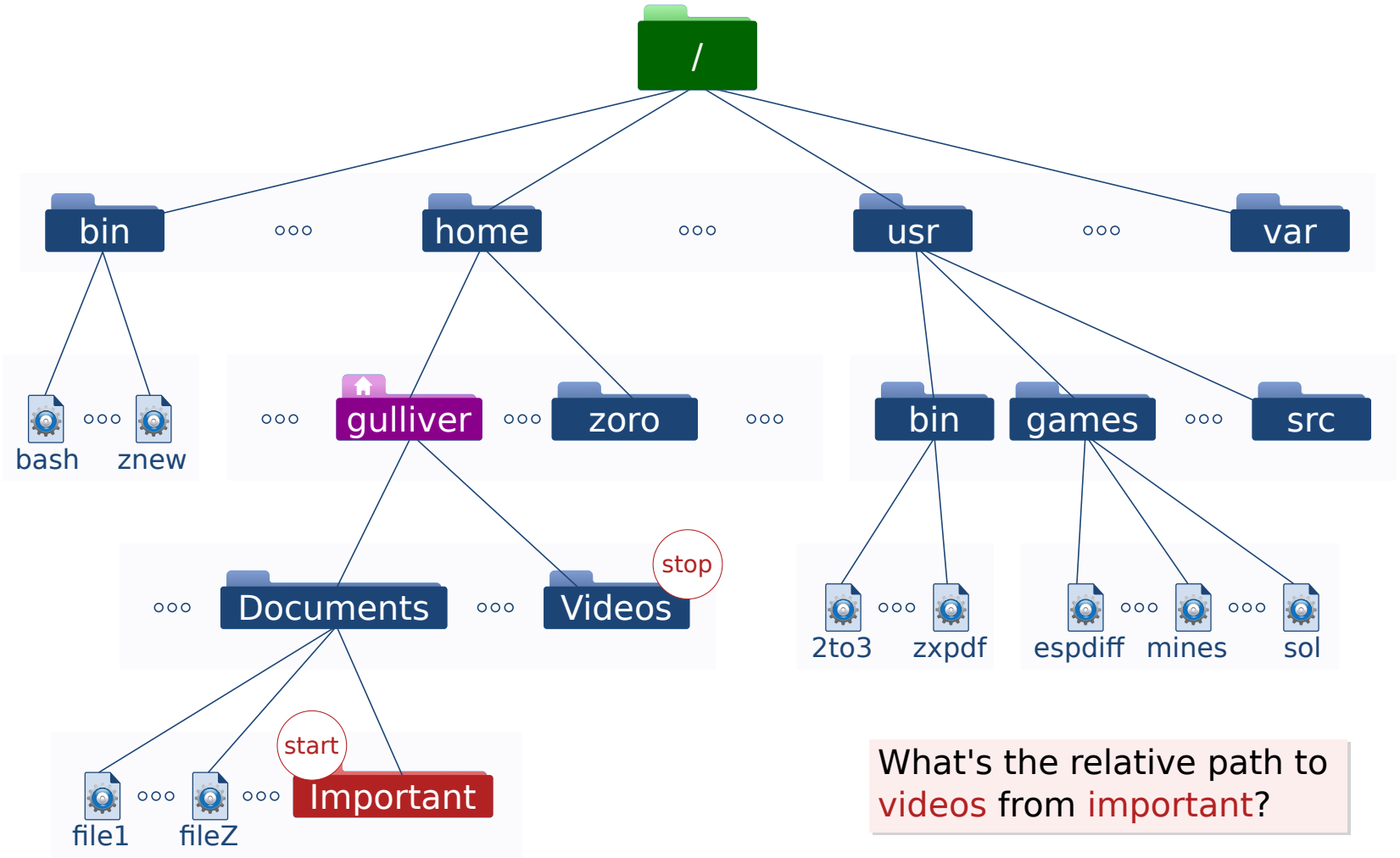
For the root folder - signalizes an absolute path.

Relative paths - start from the working directory



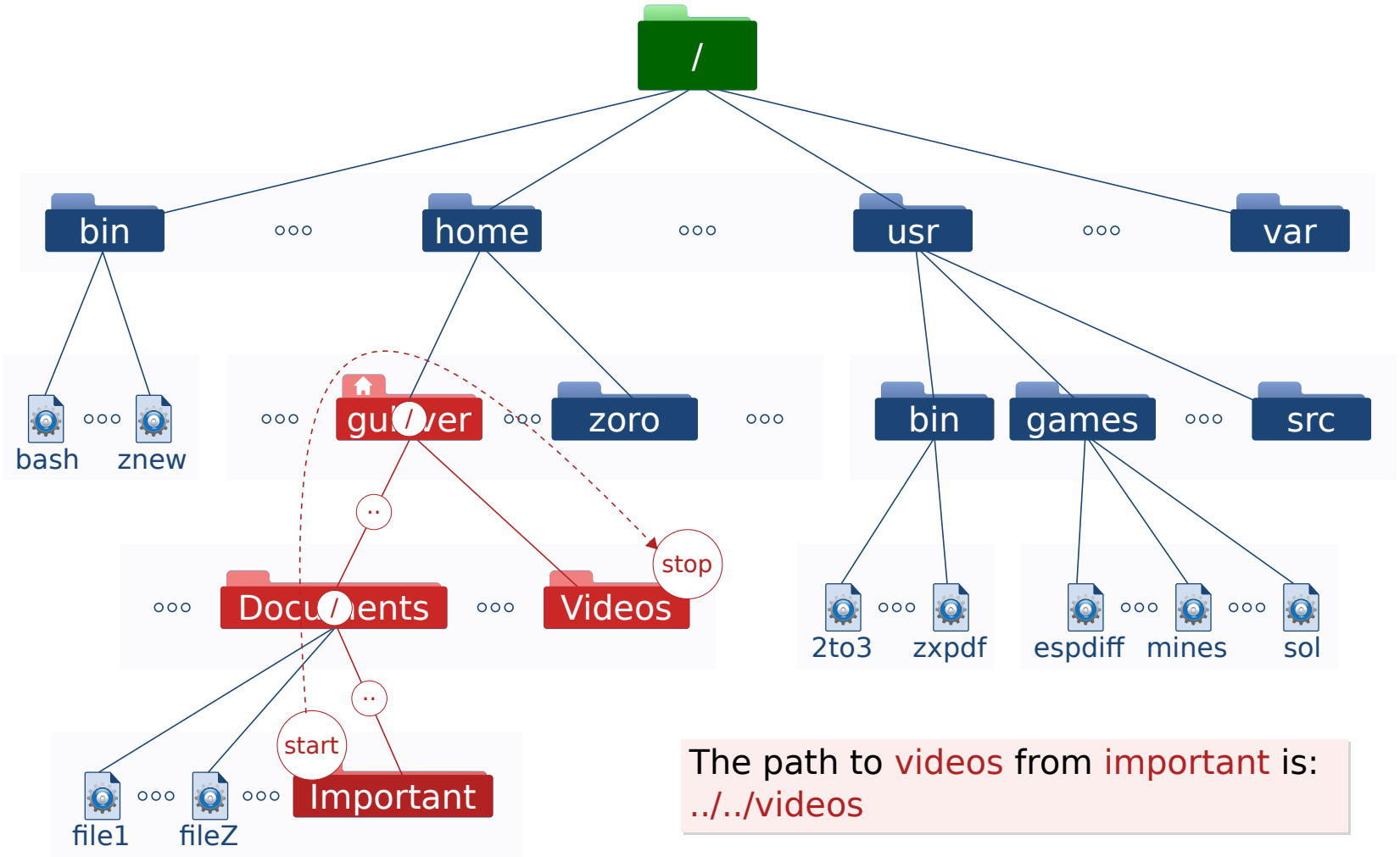
The path to mines from important is relative:
../../../../usr/games/mines

Question 1



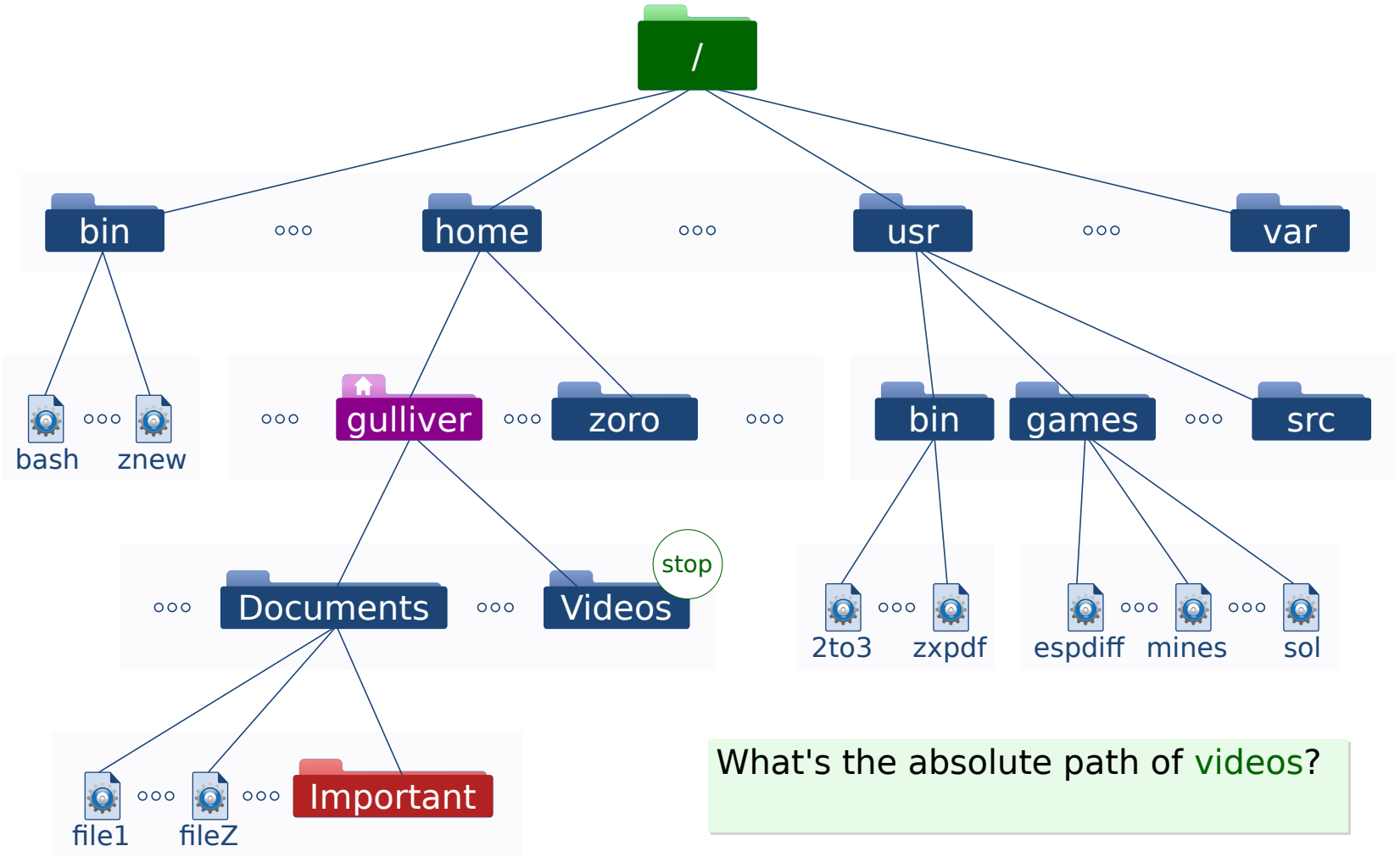
What's the relative path to videos from important?

Question 1



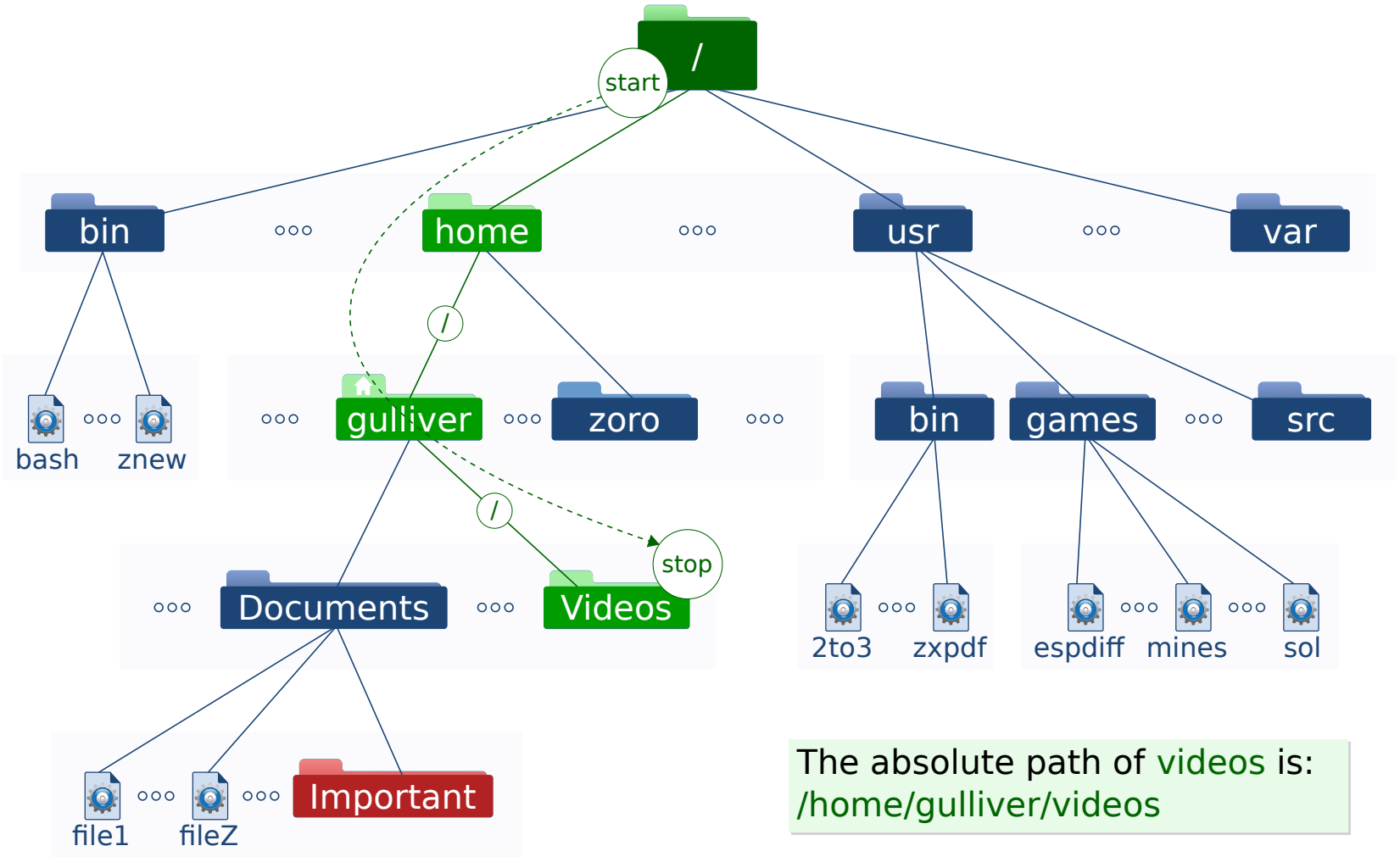
The path to videos from important is:
../../videos

Question 2



What's the absolute path of videos?

Question 2



The absolute path of videos is:
/home/gulliver/videos

Command format

```
command [option]... [argument]...
```

Most input lines have three basic elements:

- `command` - executing program (application) name (path).
- `option(s)` - modify what the command may do.
- `argument(s)` - what the command operates on.

Command format

```
command [option]... [argument]...
```

Most input lines have three basic elements:

- `command` - executing program (application) name (path).
- `option(s)` - modify what the command may do.
- `argument(s)` - what the command operates on.

Some notes:

- The `command` may be followed by one or more `options`.
- `Options` usually start with one or two dashes.
- Plenty `commands` have no `options`, no `arguments`, or neither.
- `Commands`, `options`, and `arguments` are case sensitive.

Command format

```
command [option]... [argument]...
```

Most input lines have three basic elements:

- `command` - executing program (application) name (path).
- `option(s)` - modify what the command may do.
- `argument(s)` - what the command operates on.

Some notes:

Attention to blank spaces!

Command format examples

command	[option]...	[argument]...
pwd		
cd		/usr/bin
ls	-a -l -h	/usr/bin
ls	-alh	/usr/bin
cp	-r	data yet_other_data
du	-h --total	introduction.pdf connecting.pdf core.pdf
head	-n 100	introduction.tex
less		introduction.tex
man		less

Basic Commands Overview

- Information (help) about other commands:
  - `man`.
- Navigate through the filesystem:
  - `pwd`, `ls`, `cd`.
- Manipulate files and folders:
  - `mkdir`, `mv`, `rm`, `cp`.
- Inspect files:
  - `less`, `cat`, `head`, `tail`.
- Edit files:
  - `nano`.
- Disk usage:
  - `du`.
- Search and count:
  - `locate`, `find`, `grep`, `wc`.

man

`man [command]`

Shows a user manual for the mentioned `command`.

```
$ man less
```

```
LESS(1) General Commands Manual LESS(1)
NAME
less - opposite of more
SYNOPSIS
less -?
...
```

Search for keywords by pressing slash (/).
This also works in `less` and many other programs.
Exit by pressing `q` (quit).

man

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

Lists commands whos manual page contain a certain `keyword`.

Example: list commands that let you edit something.

```
$ man -k edit
dotty (1)  - A Customizable Graph Editor
ed (1)     - line-oriented text editor
edit (1)   - execute programs via entries in the mailcap file
editor (1) - Nano ANOther editor, an enhanced free Pico clone
ex (1)     - Vi IMproved, a programmers text editor
gedit (1)  - text editor for the GNOME Desktop
kate (1)   - Advanced text editor for KDE
...
```

Navigate Through The Filesystem

`pwd [option]...`

Prints the absolute **p**ath of the current/**w**orking **d**irectory.

`ls [option]... [directory]...`

Lists information about the `directory` in alphanumeric order.
If no `directory` is specified the `working directory` is assumed.

`cd [directory]`

Change working directory to the provided `directory`.

Navigate Through The Filesystem

```
$
```

Navigate Through The Filesystem

Find where you are

```
$ pwd
```

Navigate Through The Filesystem

Find where you are

```
$ pwd
/home/gulliver
$
```

Navigate Through The Filesystem

Find where you are

```
$ pwd
/home/gulliver
$
```

Navigate Through The Filesystem

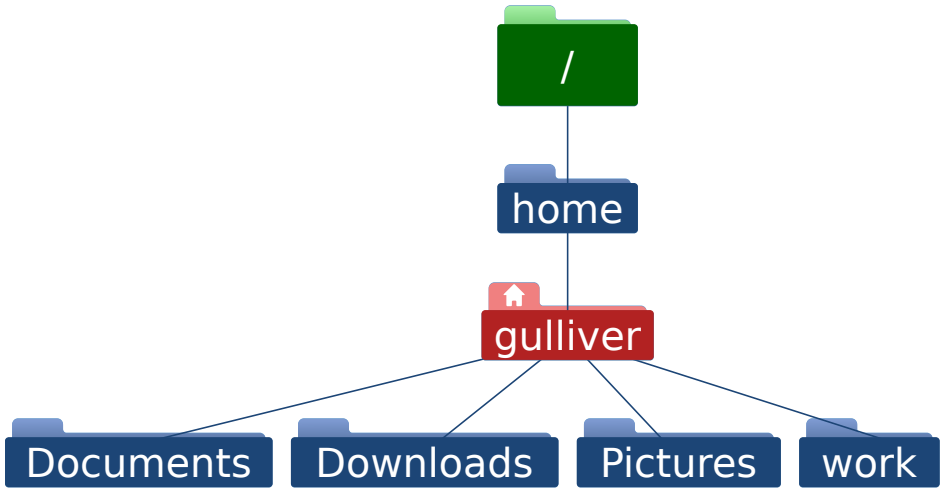
List the contents of the working directory

```
$ ls
```

Navigate Through The Filesystem

List the contents of the working directory

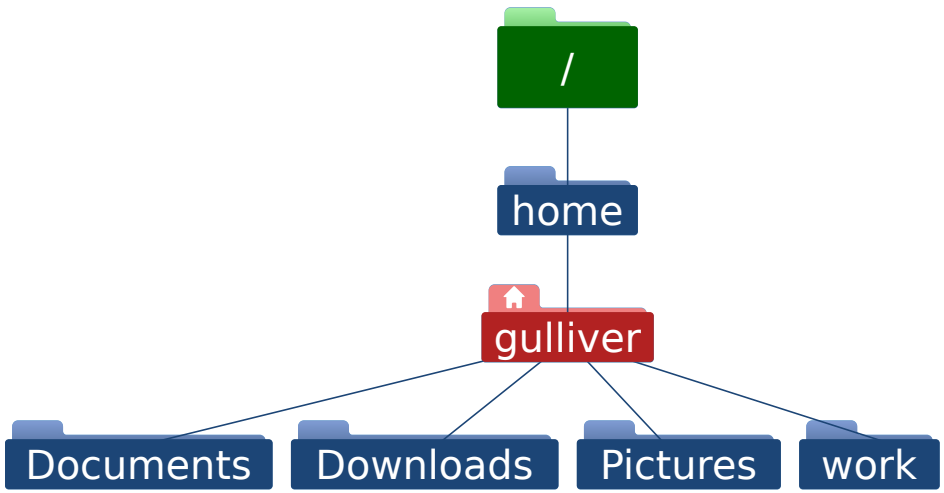
```
$ ls
Documents    Downloads    Pictures    work
$
```



Navigate Through The Filesystem

List the contents of the root directory

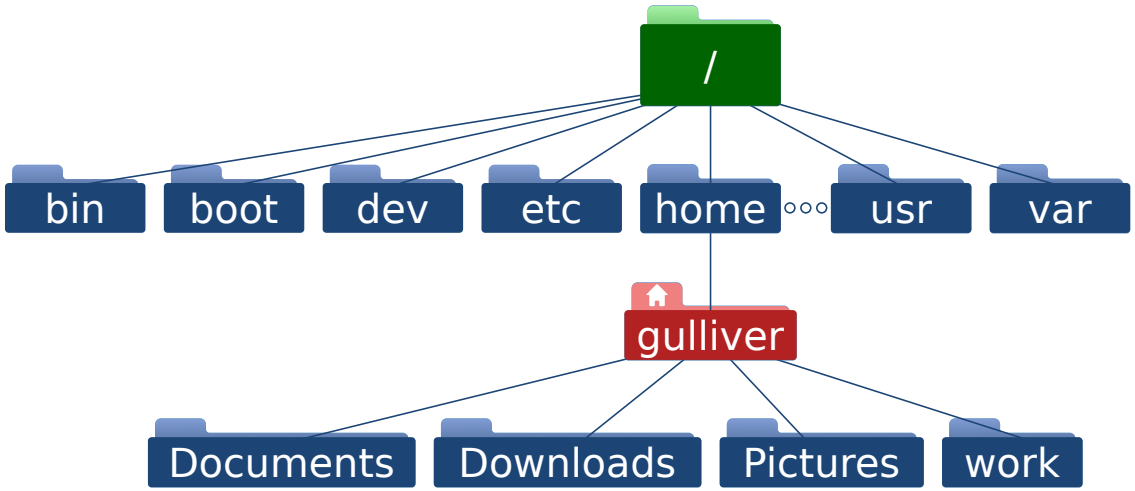
```
$ ls /
```



Navigate Through The Filesystem

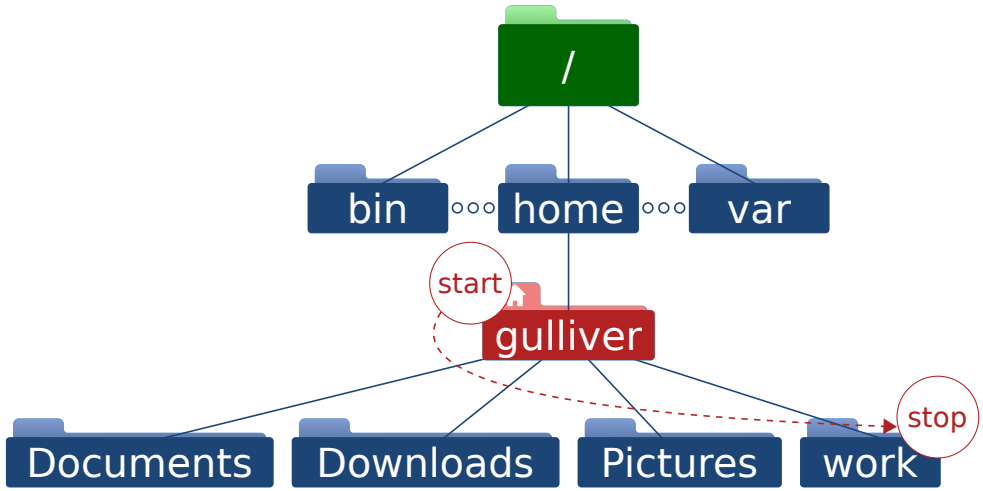
List the contents of the root directory

```
$ ls /
bin   boot   dev    etc   home   ...   usr   var
$
```



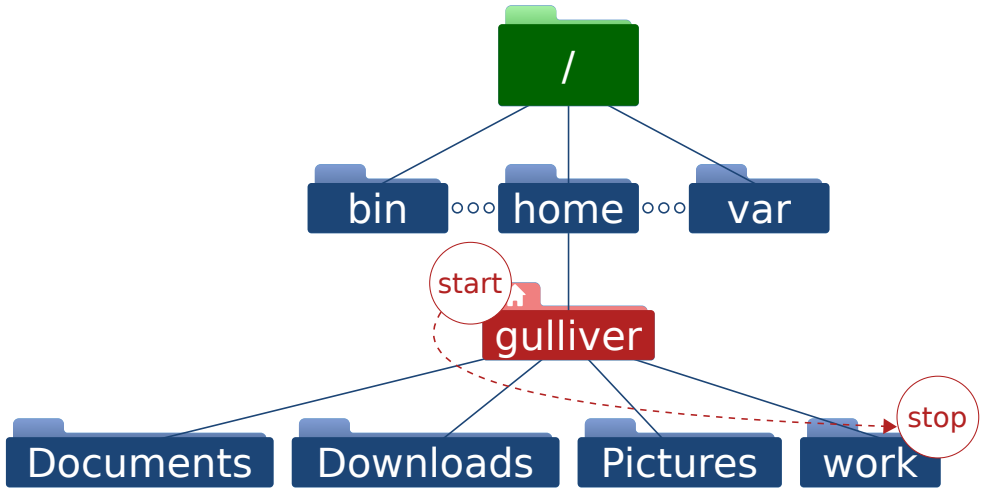
Change current directory

```
$ cd
```



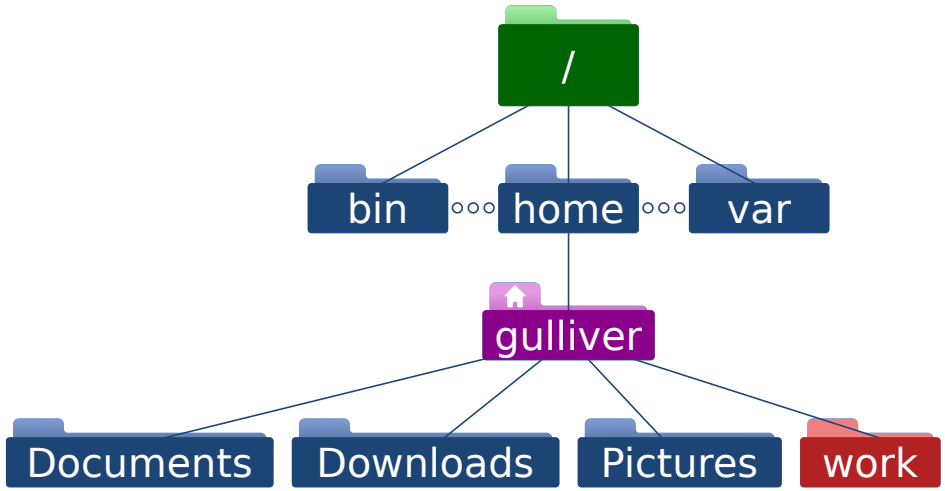
Change current directory

```
$ cd work
```



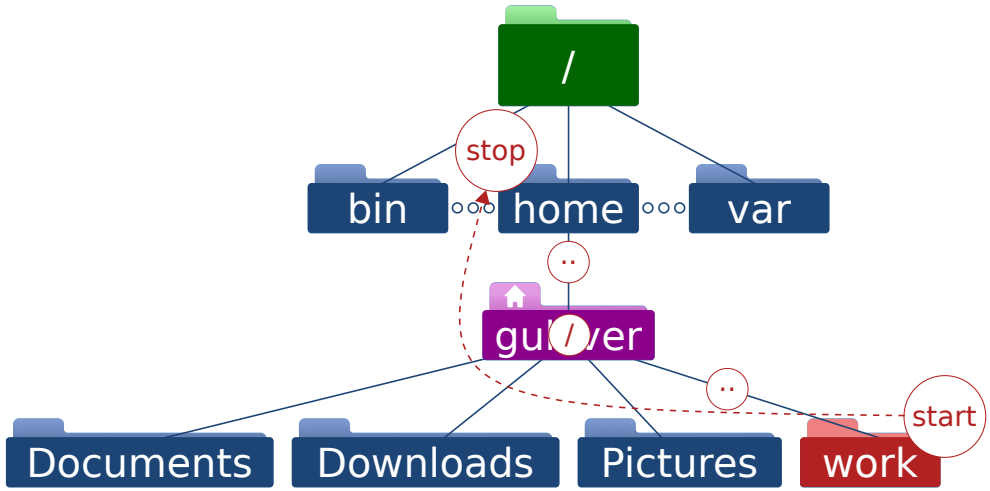
Change current directory

```
$ cd work
$
```



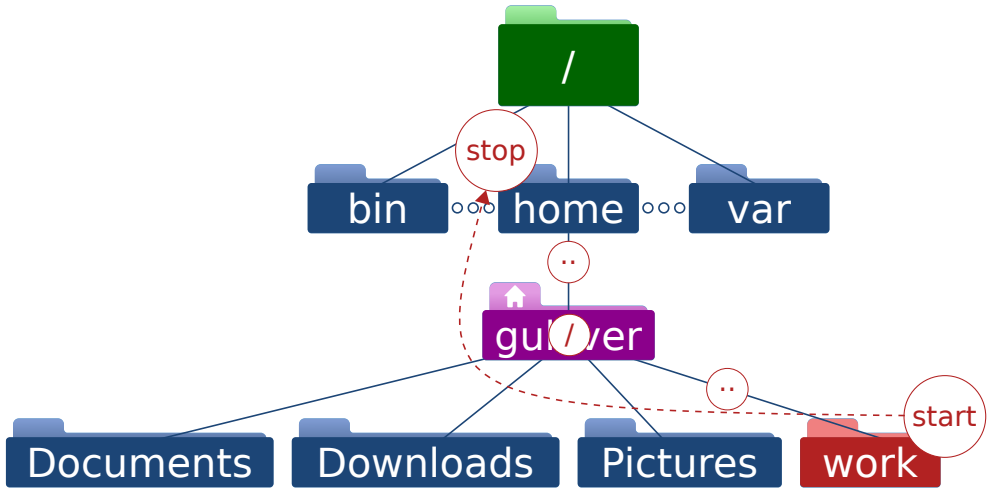
Change current directory

```
$ cd
```



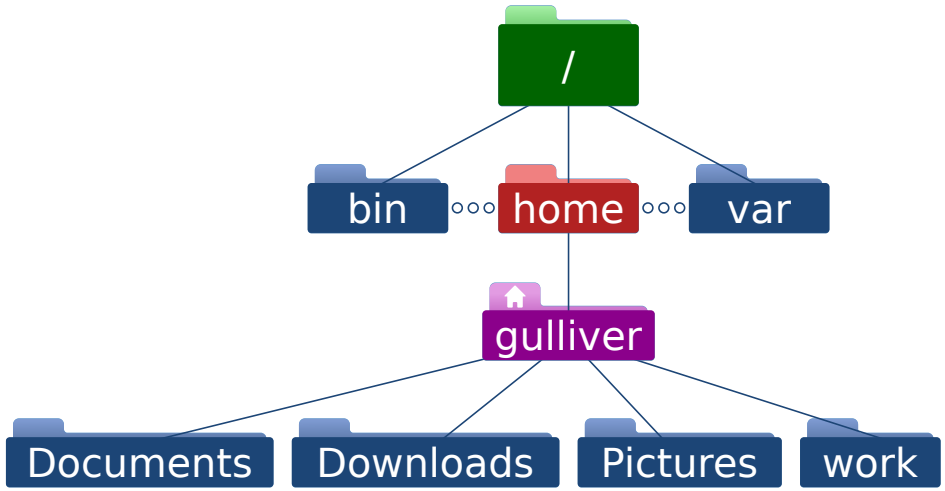
Change current directory

```
$ cd ../..
```



Change current directory

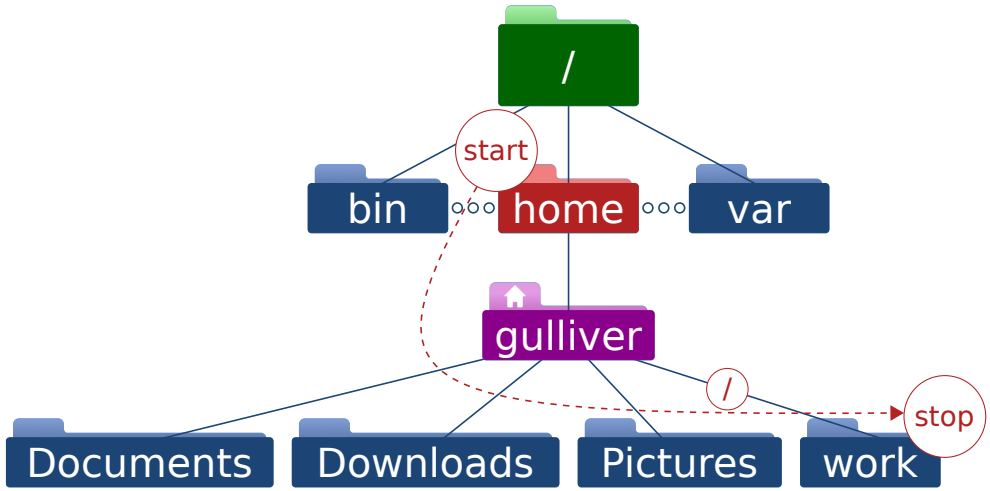
```
$ cd ../..
$
```



Navigate Through The Filesystem

List the contents of another directory

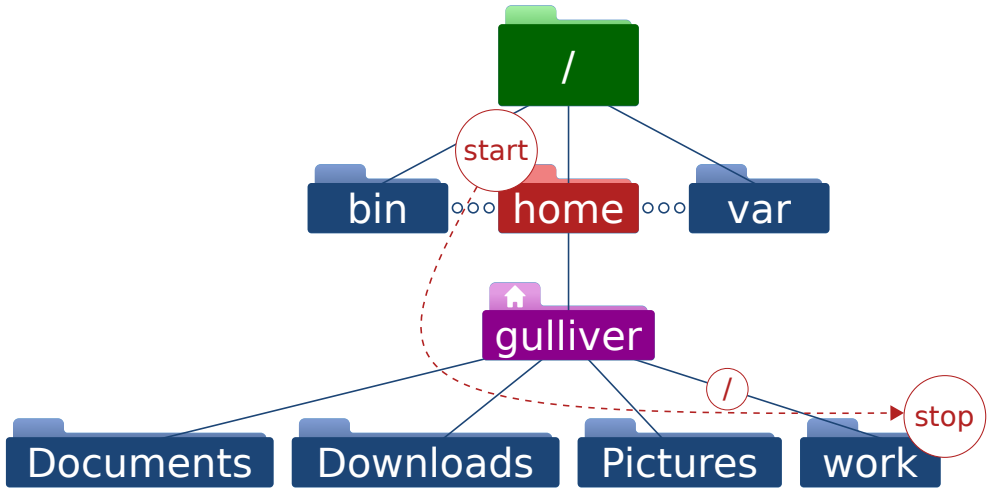
```
$ ls
```



Navigate Through The Filesystem

List the contents of another directory

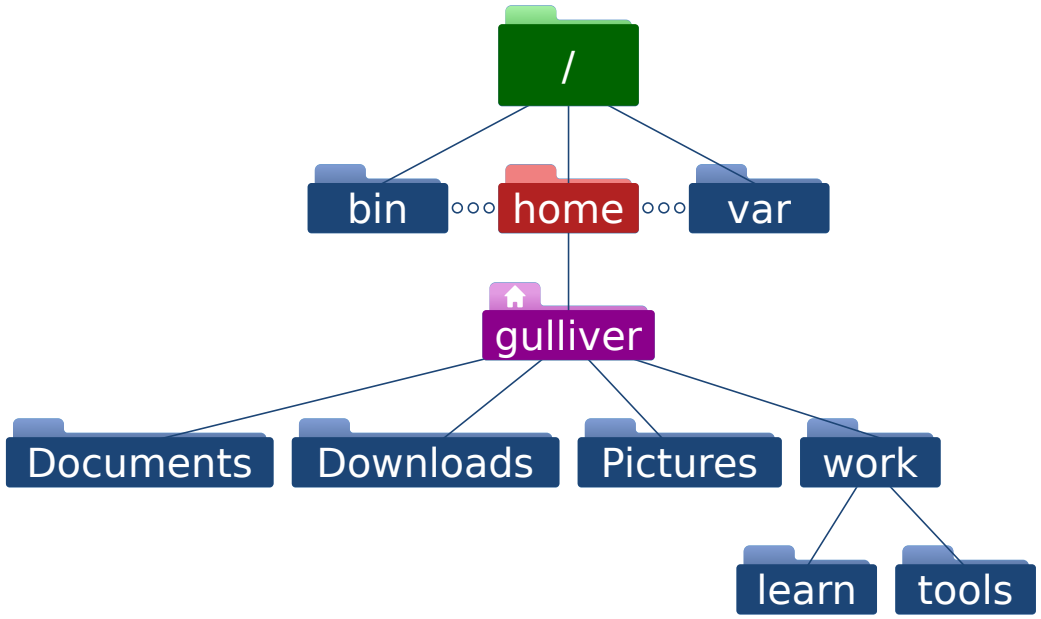
```
$ ls gulliver/work
```



Navigate Through The Filesystem

List the contents of another directory

```
$ ls gulliver/work
learn  tools
$
```



Some ls options

- `-l` (long)
  Long listing: type, permissions, owner, group, size, and date.
- `-S` (size)
  Biggest files first.
- `-h, --human-readable`
  With `-l` and/or `-s` prints human readable sizes.
- `-r, --reverse`
  Reverses the sort order.
- `-F, --classify`
  Append indicator type to entries, e.g, sufix '/' for directory.

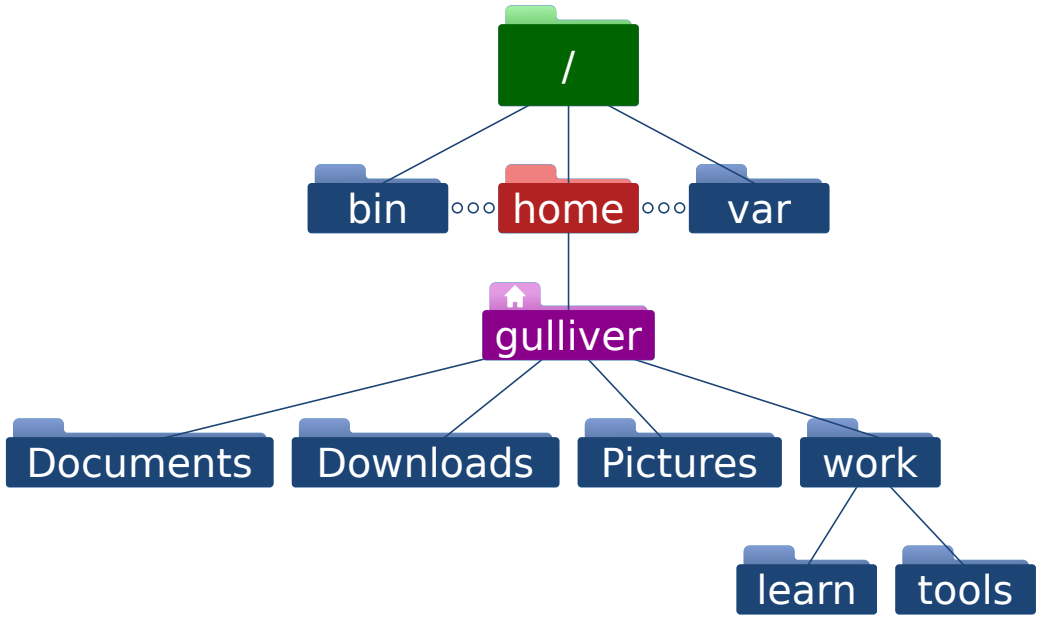
Some ls options

```
$ ls -lSh /user/games/
-rwxr-xr-x 1 root root 276K nov 12  2015 sol
-rwxr-xr-x 1 root root 158K aug 25 08:54 gnome-sudoku
-rwxr-xr-x 1 root root 109K feb 11  2016 gnome-mines
-rwxr-xr-x 1 root root 105K nov 12  2015 gnome-mahjongg
-rwxr-xr-x 1 root root 1,6K mei 16  2015 espdiff
```

Some ls options

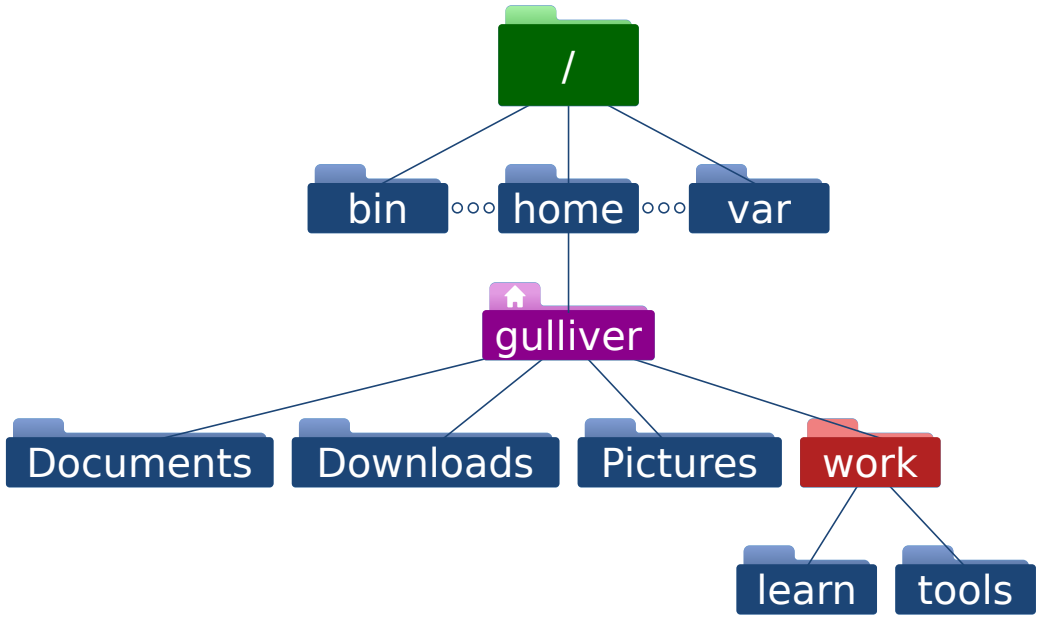
Change directory shortcuts - go to previous working directory

```
$ cd -
```



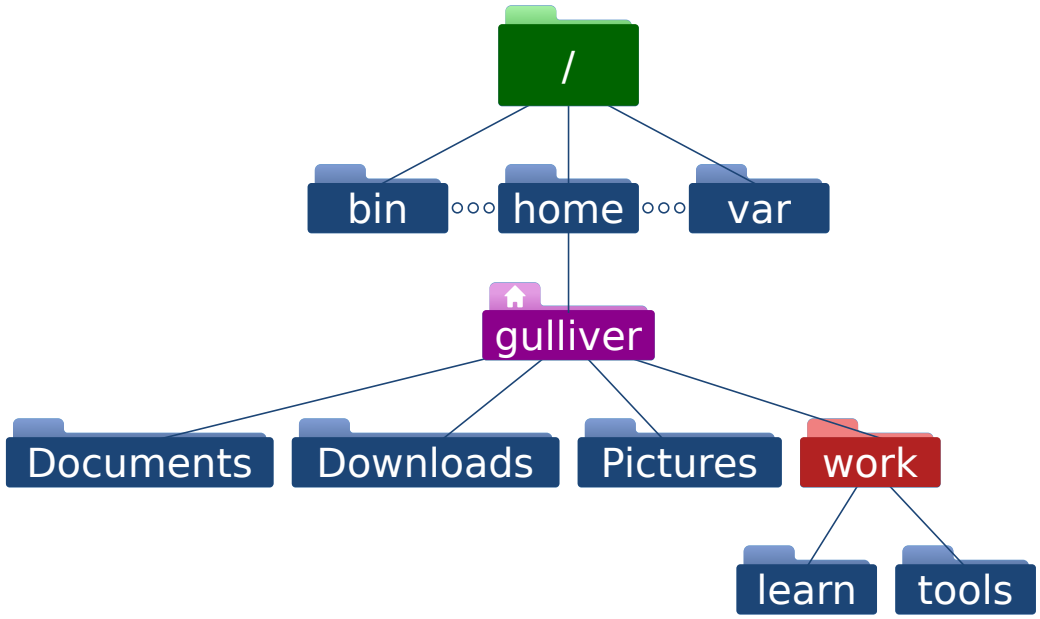
Change directory shortcuts - go to previous working directory

```
$ cd -
$
```



Change directory shortcuts - refer to your home directory

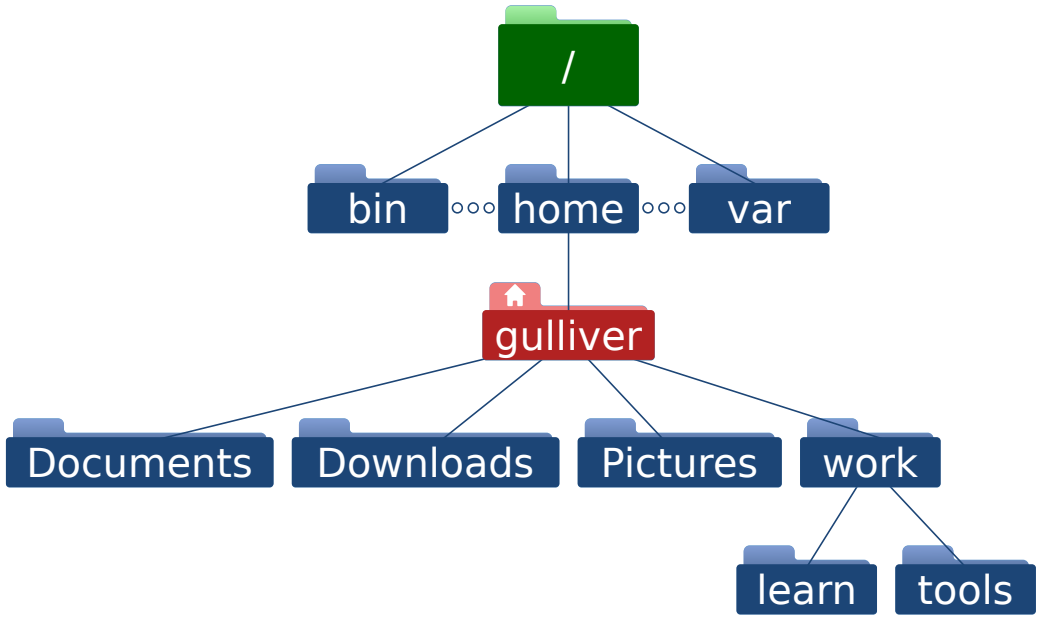
```
$ cd ~
```



Navigate Through The Filesystem

Change directory shortcuts - refer to your home directory

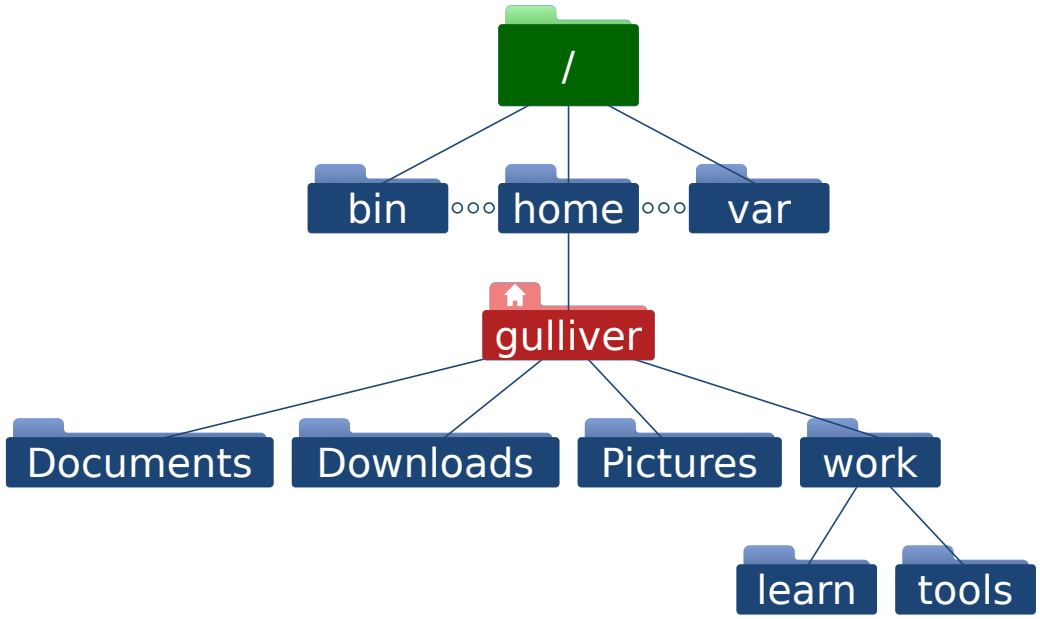
```
$ cd ~
$
```



Navigate Through The Filesystem

Change directory shortcuts - refer to your home directory

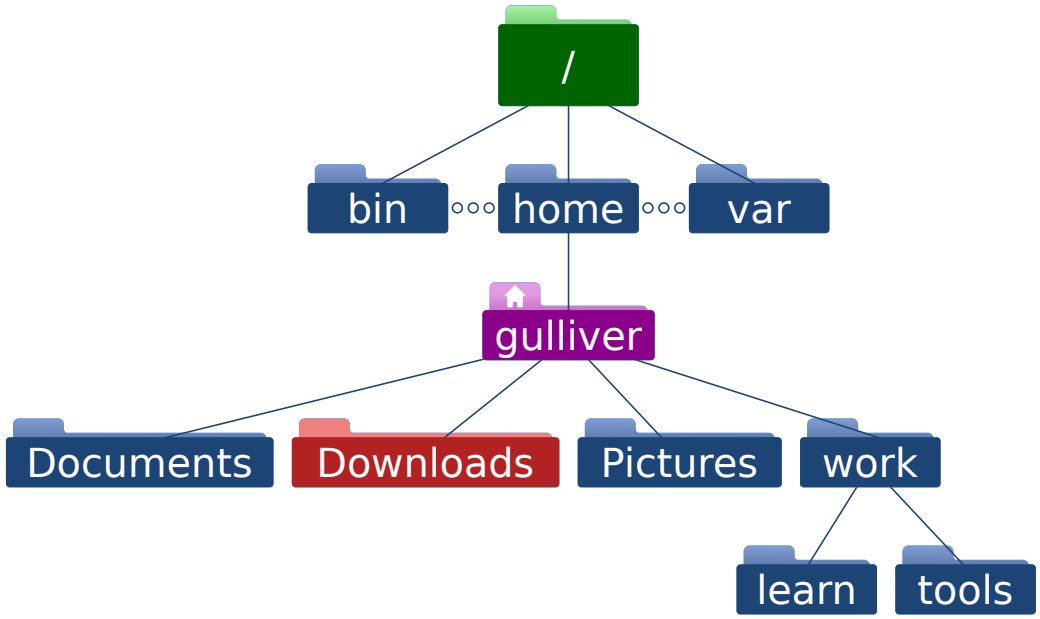
```
$ cd ~/Downloads
```



Navigate Through The Filesystem

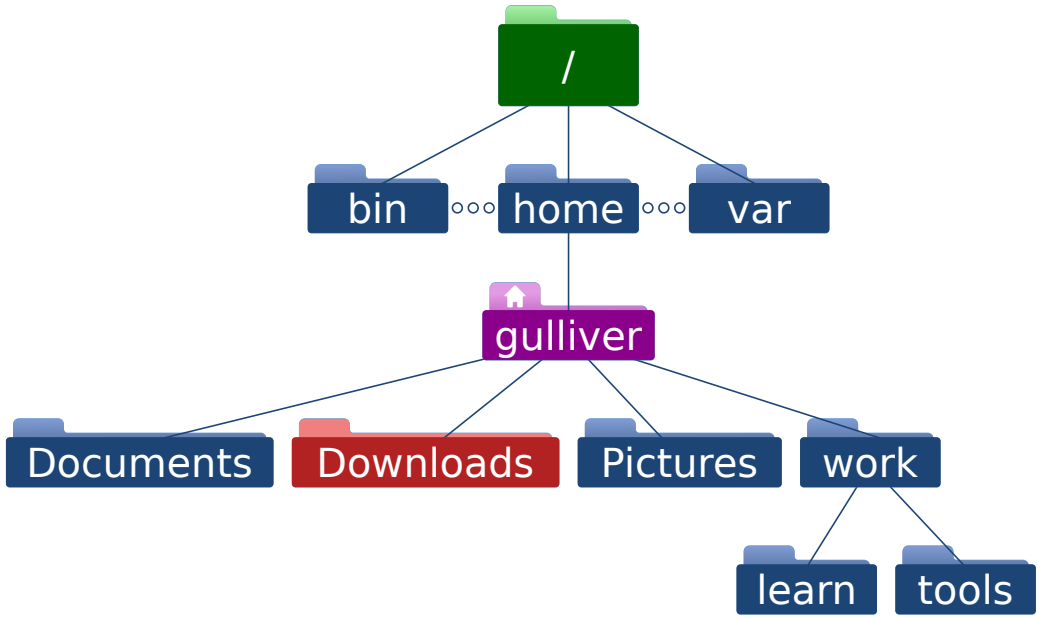
Change directory shortcuts - refer to your home directory

```
$ cd ~/Downloads
$
```



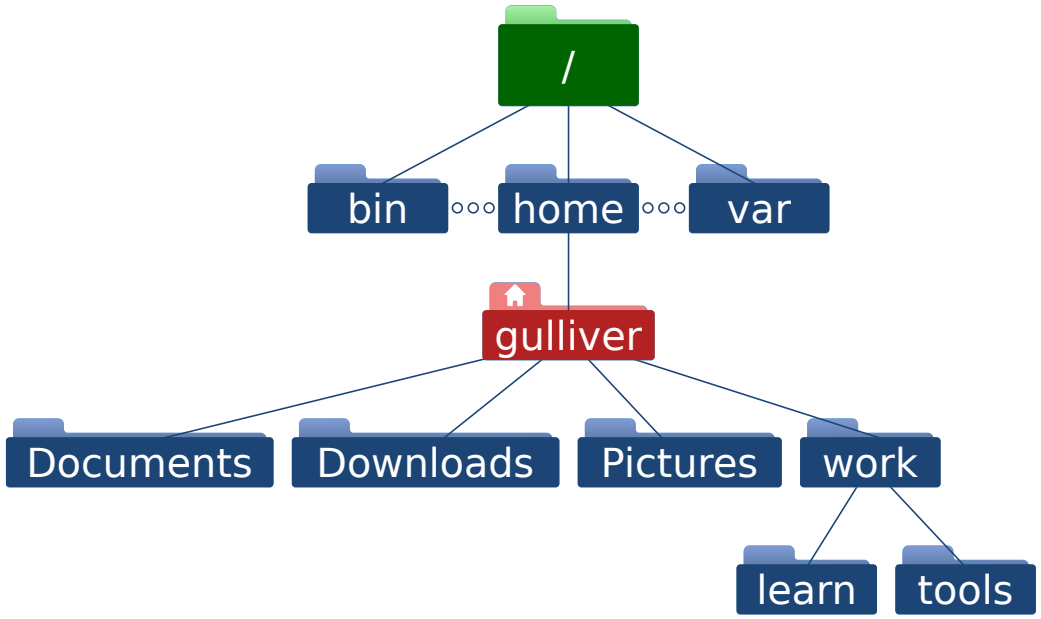
Change directory - no argument

```
$ cd
```



Change directory - no argument

```
$ cd
$
```



Question 3

Starting from `/home/antonia/important/`, which of the following commands could Antonia use to navigate to her home directory, which is `/home/antonia`?

1. cd /
2. cd /home/antonia
3. cd ../..
4. cd ~
5. cd home
6. cd ~/important/..
7. cd
8. cd ..

Question 3

Starting from `/home/antonia/important/`, which of the following commands could Antonia use to navigate to her home directory, which is `/home/antonia`?

1. cd /
2. cd /home/antonia
3. cd ../..
4. cd ~
5. cd home
6. cd ~/important/..
7. cd
8. cd ..

Check executed commands

`history`

Shows the list of previously executed commands.

```
$ history LESS(1) General Commands Manual LESS(1)
2026 ls
2027 ls gulliver/work
2028 cd -
2029 cd
2030 cd   /Downloads
```

Summary

- User - terminal - shell interaction.
- File system tree.
- Command line interface.
- File system tree navigation.

Leiden University
Medical Center

Acknowledgements

Mihai Lefter
Jonathan Vis
Jeroen Laros

Extras

Find out default shell:

```
$ ps -p $$
  PID TTY          TIME CMD
24421 pts/4    00:00:00 bash
$ echo "$SHELL"
/bin/bash
$ echo "$0"
bash
```

Find out your username:

```
$ whoami
gulliver
```