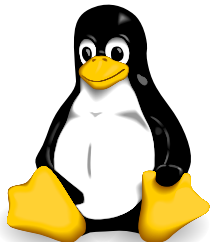


Introduction to Linux



scitas.epfl.ch

May 14, 2018

[http://go.epfl.ch/
intro-linux-ex](http://go.epfl.ch/intro-linux-ex)

What is Linux

Just for fun

- 1991 Linus Torvalds starts Linux “for fun”
- *“I’m doing a (free) operating system (just a hobby, won’t be big and professional like gnu) for 386(486) AT clones.”*

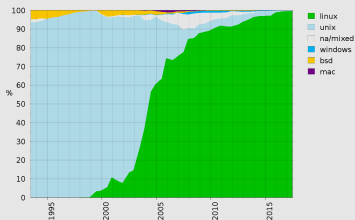


What is a Linux distribution

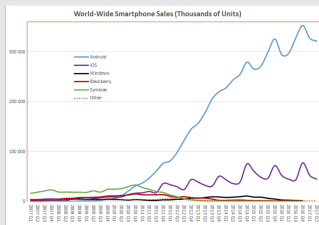
- Linux Kernel + softwares
- Desktop distributions: windowing system (X11) + desktop manager (GNOME, KDE), package manager, service manager
- Common distributions: Debian, Ubuntu, Redhat, Arch, ...

Where can you find Linux

Supercomputers



Smartphones



Other numbers

- Desktop: 3.37%
- Servers (web): 37%
- Servers (mainframe): 28%
- Embedded: 29.44%

Login

First step: get a shell

- Shell: user interface to the operating system's services
- Start a shell on your local machine
- Connect to a remote machine to get a shell
- SSH: Secure SHell

Windows

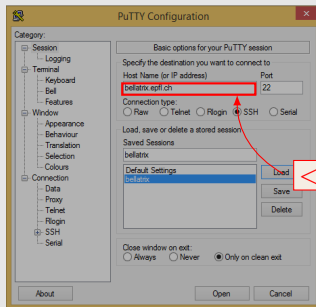
- Git Bash
- Putty
- Tunnelier

Unix like

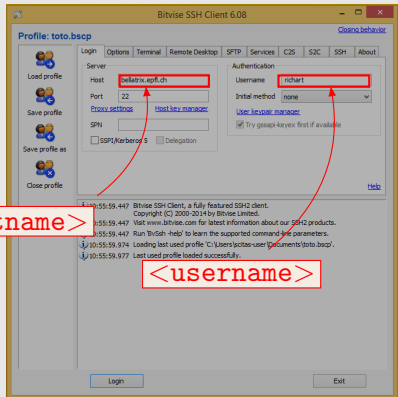
- Start a terminal

Login: Windows

Putty

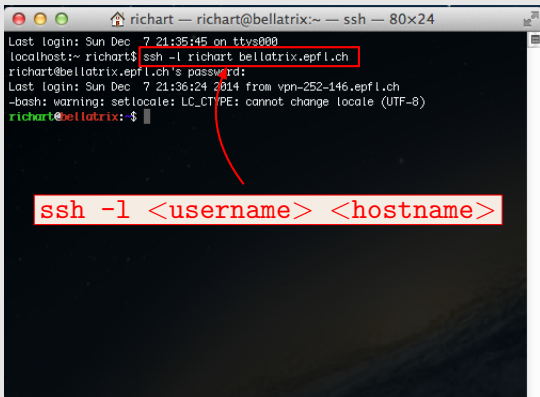


Tunnelier



Login: Mac OS

Terminal



```
richart — richart@bellatrix:~ — ssh — 80x24
Last login: Sun Dec 7 21:35:45 on ttys000
localhost:~ richart$ ssh -l richart bellatrix.epfl.ch
richart@bellatrix.epfl.ch's password:
Last login: Sun Dec 7 21:36:24 2014 from vpn-252-146.epfl.ch
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8)
richart@bellatrix:~$
```

`ssh -l <username> <hostname>`

Some usefull commands

ls

list files

cd

change directory

pwd

path of the current directory

man

get help on commands

apropos

search for a command and print a brief description

echo

print the arguments in the terminal

Exercise 1: First commands

Questions:

- Connect to the remote machine `scitascours1.epfl.ch`.
- Check your current folder and your files.
- Check the help of `ls` (Q to quit).

Exercise 1: First commands

Questions:

- Connect to the remote machine `scitascours1.epfl.ch`.
- Check your current folder and your files.
- Check the help of `ls` (Q to quit).

```
> pwd  
> ls
```

Exercise 1: First commands

Questions:

- Connect to the remote machine `scitascours1.epfl.ch`.
- Check your current folder and your files.
- Check the help of `ls` (Q to quit).

```
> pwd  
> ls
```

```
> man ls
```

File structure

“On UNIX everything is a file”

Directories	files that are lists of other files
Special files	files for input/output mechanisms
Links	way to make files visible in different parts of the file system
Sockets, named pipes	way to communicate between processes

File structure

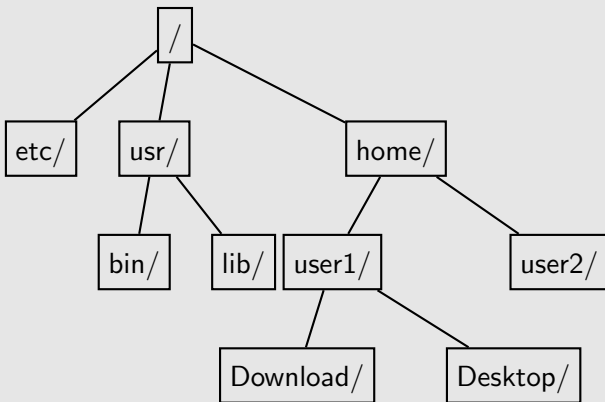
Standard directories

/etc	System configurations files
/boot	Boot files, kernel and boot loader
/home	User accounts
/usr	Software, libraries, documentations, for user programs
/var	Variables files such as log files, mail queues, databases
/opt	Third-party and extra softwares

Virtual directories

/dev	Peripheral hardware related files
/proc	Informations about processes
/sys	Informations about the kernel

Tree view



Exercise 2: Directory structure

Questions:

- Show directory structure under the root folder
- Change directory to the root folder and list the content

Exercise 2: Directory structure

Questions:

- Show directory structure under the root folder
- Change directory to the root folder and list the content

```
> ls /
```

Exercise 2: Directory structure

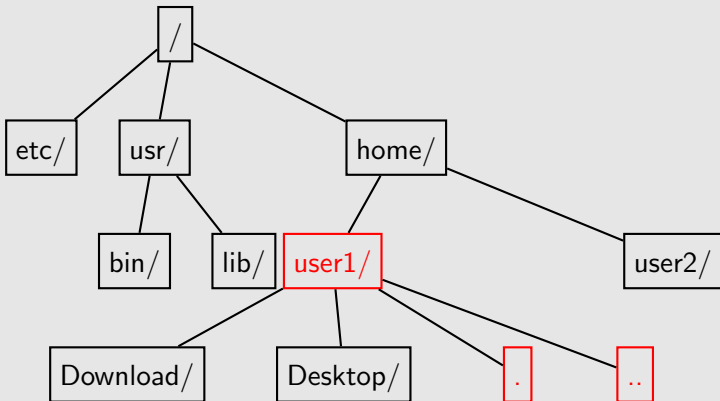
Questions:

- Show directory structure under the root folder
- Change directory to the root folder and list the content

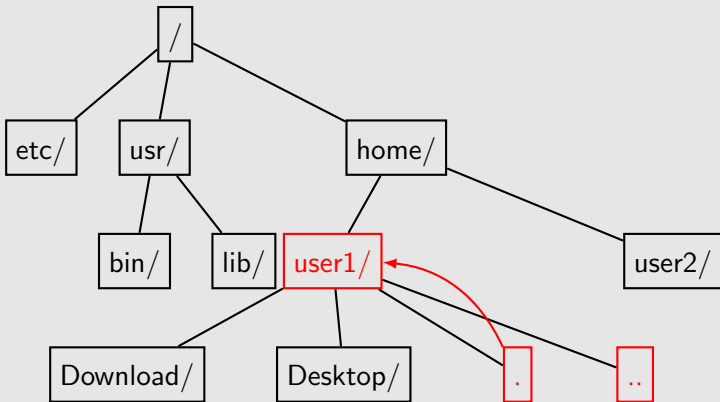
```
> ls /
```

```
> cd /  
> ls
```

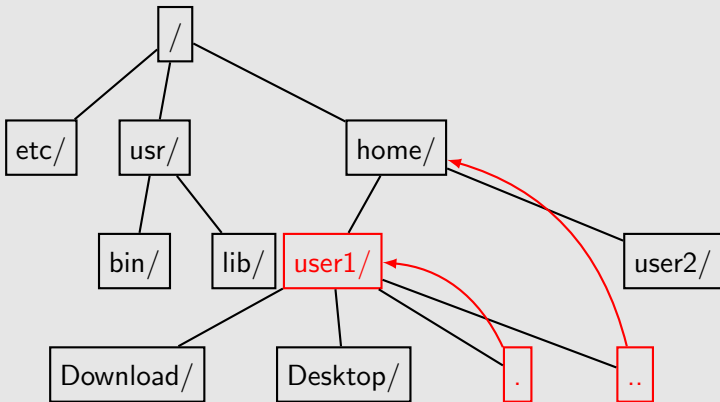

Relative/absolute path



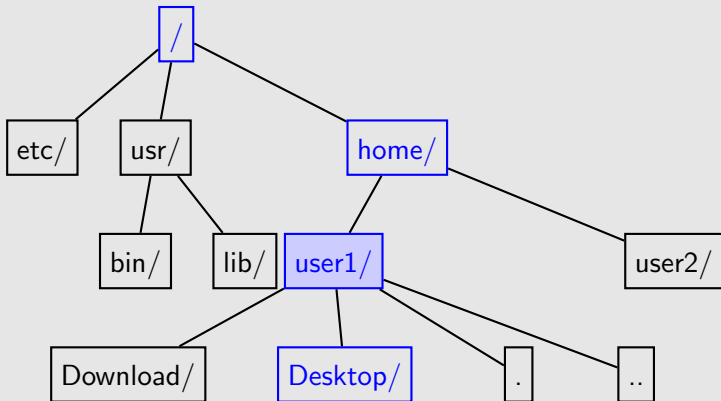
Relative/absolute path



Relative/absolute path



Relative/absolute path



`/home/user1/Desktop`

`./Desktop`

Exercise 3: Relative/absolute path

Questions:

- Try the command `ls -a`
- List the parent folder of your home
- List the root folder with a relative path
- To what correspond the folder `.././../..`

Exercise 3: Relative/absolute path

Questions:

- Try the command `ls -a`
- List the parent folder of your home
- List the root folder with a relative path
- To what correspond the folder `../../../../`

```
> ls ../
```

Exercise 3: Relative/absolute path

Questions:

- Try the command `ls -a`
- List the parent folder of your home
- List the root folder with a relative path
- To what correspond the folder `../../../../`

```
> ls ../
```

```
> ls ../../
```

Exercise 3: Relative/absolute path

Questions:

- Try the command `ls -a`
- List the parent folder of your home
- List the root folder with a relative path
- To what correspond the folder `../../../../`

```
> ls ../
```

```
> ls ../../
```

```
> cd ../../../../
> pwd
```


File creation/suppression commands

`cp`

copy files

`mv`

move/rename files

`rm`

remove file **caution no trash**

`mkdir`

create a folder

`cat`

print the content of file, used to concatenate

`touch`

update the date of a file, used to create empty files

`tail/head`

print the end/beginning of a file

`more/less`

print the content of a file "more or less the same"

`grep`

search a string in a file

`find`

find a file

`file`

the type of file based on content

Exercise 4: File handling commands

Questions:

- Test the different commands (do not forget you can check the help with `man <command>`)
- Create a directory named `exercise_linux`
- Copy the file `CodingStyle` from your home folder into the `exercise_linux` folder
- Check the content of this file
- Print the last 3 lines
- Find the lines containing the word “coding” independently of the case
- Rename the file as `KernelCodingStyle`
- Copy the folder `exercise_linux` to `exo_linux`
- Remove the folder `exercise_linux`

Answer to exercise 4

```
> mkdir exercise_linux
```

Answer to exercise 4

```
> mkdir exercise_linux
```

```
> cd exercise_linux  
> cp ../CodingStyle .
```

Answer to exercise 4

```
> mkdir exercise_linux
```

```
> cd exercise_linux  
> cp ../CodingStyle .
```

```
> less CodingStyle
```

Answer to exercise 4

```
> mkdir exercise_linux
```

```
> cd exercise_linux  
> cp ../CodingStyle .
```

```
> less CodingStyle
```

```
> tail -n3 CodingStyle
```

Answer to exercise 4

```
> mkdir exercise_linux
```

```
> cd exercise_linux  
> cp ../CodingStyle .
```

```
> less CodingStyle
```

```
> tail -n3 CodingStyle
```

```
> grep -n --color -i coding CodingStyle
```

Answer to exercise 4

```
> mkdir exercise_linux
```

```
> cd exercise_linux  
> cp ../CodingStyle .
```

```
> less CodingStyle
```

```
> tail -n3 CodingStyle
```

```
> grep -n --color -i coding CodingStyle
```

```
> mv CodingStyle KernelCodingStyle
```


Answer to exercise 4

```
> mkdir exercise_linux
```

```
> cd exercise_linux  
> cp ../CodingStyle .
```

```
> less CodingStyle
```

```
> tail -n3 CodingStyle
```

```
> grep -n --color -i coding CodingStyle
```

```
> mv CodingStyle KernelCodingStyle
```

```
> cd ..  
> cp -r exercise_linux exo_linux
```

Answer to exercise 4

```
> mkdir exercise_linux
```

```
> cd exercise_linux  
> cp ../CodingStyle .
```

```
> less CodingStyle
```

```
> tail -n3 CodingStyle
```

```
> grep -n --color -i coding CodingStyle
```

```
> mv CodingStyle KernelCodingStyle
```

```
> cd ..  
> cp -r exercise_linux exo_linux
```

```
> rm -r exercise_linux
```

File permissions

Possible permissions

Partial results of the command `ls -l CodingStyle`

```
-rw- r-- r-- richart scitas-ge CodingStyle
```

u User permissions

g Group permissions

o Other users permissions

rx/--- explanations

<code>-</code> or 0	Permission not granted
<code>r</code> or 4	Read/List permission
<code>w</code> or 2	Write/Create permission
<code>x</code> or 1	Execute/Traverse permission

Permission commands

<code>chmod</code>	change permissions
<code>chgrp</code>	change main group
<code>chown</code>	change ownership
<code>newgrp</code>	login with a new group as main group
<code>id</code>	print user and group ids

chmod details

```
chmod [ugoa][+--]=[rwxXst] files
```

```
chmod octal files
```

```
example: u=rw,go=r equivalent to 644
```

Exercise 5: Permissions

Questions:

- Print your ids
- Check the permissions of your home folders
- Change the the default group of the file `KernelCodingStyle` to `users`
- Remove the list permission on `exo_linux`
- Try to list the `exo_linux` folder of someone that as completed the previous step

Answer to exercise 5

```
> id
```

Answer to exercise 5

```
> id
```

```
> ls -l
```

Answer to exercise 5

```
> id
```

```
> ls -l
```

```
> chgrp users exo_linux/KernelCodingStyle
```


Answer to exercise 5

```
> id
```

```
> ls -l
```

```
> chgrp users exo_linux/KernelCodingStyle
```

```
> chmod go-r exo_linux
```

Answer to exercise 5

```
> id
```

```
> ls -l
```

```
> chgrp users exo_linux/KernelCodingStyle
```

```
> chmod go-r exo_linux
```

```
> ls -l /home/richart/exo_linux
```

```
> ls -l /home/richart/exo_linux/KernelCodingStyle
```

Permissions: the special case of root

The root account

- It's a special account, to which regular permissions do not apply
- Used to administer the system
- Should only be used to install trusted software
- Preferably via the `sudo` command

More about Shell

- Different shells: sh, bash, csh, tcsh, zsh, etc.
- For EPFL account: <https://cadiwww.epfl.ch/cgi-bin/accountprefs/>
- In general `echo $SHELL`

What is happening when you run a command

- Expansions
- Redirection
- Execution

Key shortcuts

Tab	auto-complete
(up)	go back in the command history
(down)	go down in the command history
Ctrl-r	reverse search in the history
Ctrl-d	exit the shell

Expansions

- Accolade: `{a{b,c},d}` → `ab ac d`
 - Tilde: `~richart/` → `/home/richart`
 - Variables: `export FOO=bar` `$FOO` → `bar`
 - Command: `$(command)` → result of command
 - Split spaces: `arg1 arg2 arg3` the quoted arguments are not split
 - Path: `*` matches any string
`?` matches any single character
`[...]` matches any character in the brackets
- **TIP:** use `echo <command>` to test your expansions

Exercise 6: Expansions

Questions:

- Try the command `touch file{0,1,2}{0,1,2,3,4,5}.log`
- List your home folder with a tilde expression
- Set a variable `FOO` to the content of your choice and print its content
- Try the commands:

```
touch foo bar
touch "foo bar"
export BROL=brol
touch "foo bar $BROL"
touch 'foo bar $BROL'
```
- List all file of the form `file<number>.log`
- List the files `file<number>.log` where `<number>` is any number with last digit 2,3 or 4
- Remove the previous files using the results of the ls command

Answer to exercise 6

```
> ls ~/
```

Answer to exercise 6

```
> ls ~/
```

```
> export FOO="the content of my choice"  
> echo $FOO
```


Answer to exercise 6

```
> ls ~/
```

```
> export FOO="the content of my choice"  
> echo $FOO
```

The second command create a file name containing a space that can be escaped with `\`

```
> rm foo\ bar foo bar foo\ bar\ \ $BROL foo\ bar\ bro1
```

Answer to exercise 6

```
> ls ~/
```

```
> export FOO="the content of my choice"  
> echo $FOO
```

The second command create a file name containing a space that can be escaped with `\`

```
> rm foo\ bar foo bar foo\ bar\ \ $BROL foo\ bar\ bro1
```

```
> ls file*.log
```

Answer to exercise 6

```
> ls ~/
```

```
> export FOO="the content of my choice"  
> echo $FOO
```

The second command create a file name containing a space that can be escaped with `\`

```
> rm foo\ bar foo bar foo\ bar\ \ $BROL foo\ bar\ bro1
```

```
> ls file*.log
```

```
> ls file{2..4}.log
```

Answer to exercise 6

```
> ls ~/
```

```
> export FOO="the content of my choice"  
> echo $FOO
```

The second command create a file name containing a space that can be escaped with `\`

```
> rm foo\ bar foo bar foo\ bar\ \ $BROL foo\ bar\ brol
```

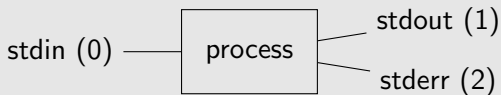
```
> ls file*.log
```

```
> ls file{2..4}.log
```

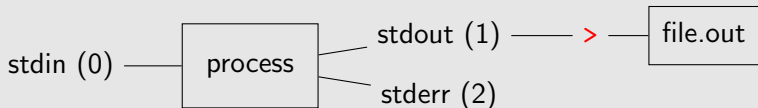
```
> rm $(ls file*.log)
```

Redirection

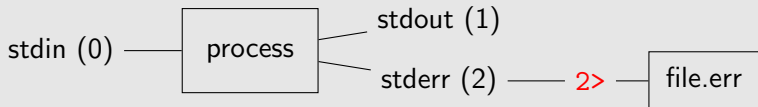
`process`



`process > file.out`

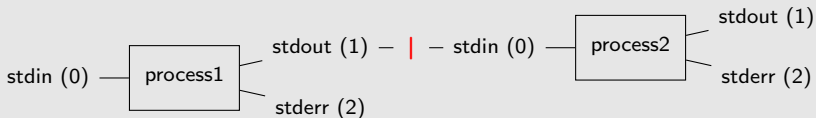


`process 2> file.err`

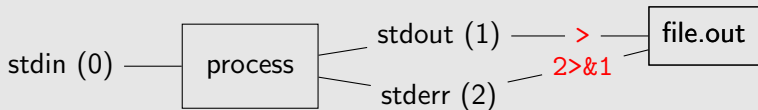


Redirection

process1 | process2



process > file.out 2>&1



Exercise 7: Redirection

Questions:

- Redirect the content of the `date` command in the `date.log` file
- Count the number of words generated by the command `date` with the help of the command `wc`
- Redirect the error output for the command `ls / tmp`

Exercise 7: Redirection

Questions:

- Redirect the content of the `date` command in the `date.log` file
- Count the number of words generated by the command `date` with the help of the command `wc`
- Redirect the error output for the command `ls / tmp`

```
> date > date.log
```


Exercise 7: Redirection

Questions:

- Redirect the content of the `date` command in the `date.log` file
- Count the number of words generated by the command `date` with the help of the command `wc`
- Redirect the error output for the command `ls / tmp`

```
> date > date.log
```

```
> date | wc -w
```

Exercise 7: Redirection

Questions:

- Redirect the content of the `date` command in the `date.log` file
- Count the number of words generated by the command `date` with the help of the command `wc`
- Redirect the error output for the command `ls / tmp`

```
> date > date.log
```

```
> date | wc -w
```

```
> ls / tmp 2> errors
```

Execution

- Execution generate a new process
- A process is identified by a pid
- Can be run in foreground or background (`&`)

Commands

`ps`

list the running processes

`jobs`

list the jobs running in the current shell

`kill`

kill processes

`bg`

resume jobs in background

`fg`

resume jobs in foreground

Key shortcuts

`Ctrl-c`

kill the current job in foreground

`Ctrl-z`

suspend the current job in foreground

Exercise 8: Execution

Questions:

- Run the command `(sleep 30; echo Slept well)`
- Run the command `(sleep 30; echo Slept well)&`
- Now you have 30 seconds to check the pid with `ps`
- If you still have spare seconds kill the process before it print on screen

Command line editors: nano vi emacs

nano

- `Ctrl-o` to save file
- `Ctrl-x` to quit

vi

- command mode and edition mode
- `i` to insert, `esc` to go back to command mode
- `:wq` to save and quit
- `:q!` to force quit

emacs

- `Ctrl-x Ctrl-s` to save file
- `Ctrl-x Ctrl-c` to quit

Exercise 9: Execution

Questions:

- Try this editors
 - nano
 - vim
 - emacs

Shell scripts

What is it

- It is a file containing a list of commands to execute in order
- Can contain loops (for, while), conditions (if)
- Often start with interpreter information `#!/bin/bash`

Example of bash script

- `~/ .bashrc`
- cluster job script

Exercise 10: Shell scripts

Questions:

- Write a script that print Hello World, wait 2seconds and then print the date
- Make your script executable
- Run it

Exercise 10: Shell scripts

Questions:

- Write a script that print Hello World, wait 2seconds and then print the date
- Make your script executable
- Run it

hello.sh

```
#!/bin/bash  
echo Hello World  
sleep 2  
date
```

Exercise 10: Shell scripts

Questions:

- Write a script that print Hello World, wait 2seconds and then print the date
- Make your script executable
- Run it

hello.sh

```
#!/bin/bash  
echo Hello World  
sleep 2  
date
```

```
> chmod +x ./hello.sh
```

Exercise 10: Shell scripts

Questions:

- Write a script that print Hello World, wait 2seconds and then print the date
- Make your script executable
- Run it

hello.sh

```
#!/bin/bash  
echo Hello World  
sleep 2  
date
```

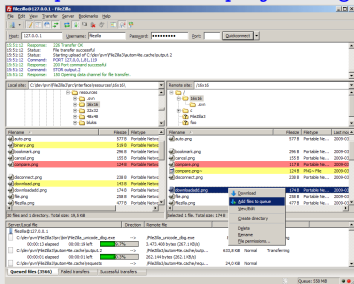
```
> chmod +x ./hello.sh
```

```
> ./hello.sh
```

Transferring files: SFTP clients

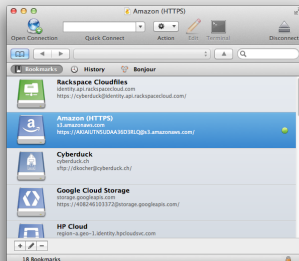
Filezilla

<https://filezilla-project.org>



Cyberduck

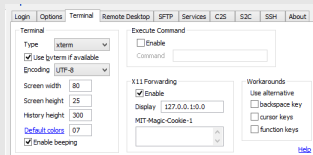
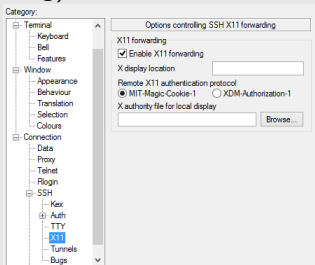
<https://cyberduck.io/>



X11 Forwarding

Windows

Install Xming <http://sourceforge.net/projects/xming/> or XWin32 (distrilog)



Mac OS

Install XQuartz <http://xquartz.macosforge.org/landing/>, MacPort or Homebrew

Connect with `ssh -Y <username>@<hostname>`

Sources

- Wikipedia: <http://en.wikipedia.org/>
 - [History_of_Linux](#)
 - [Usage_share_of_operating_systems](#)
 - [Mobile_operating_system](#)
- DistroWatch: <http://distrowatch.com/>
- [The Linux Documentation Project](#)