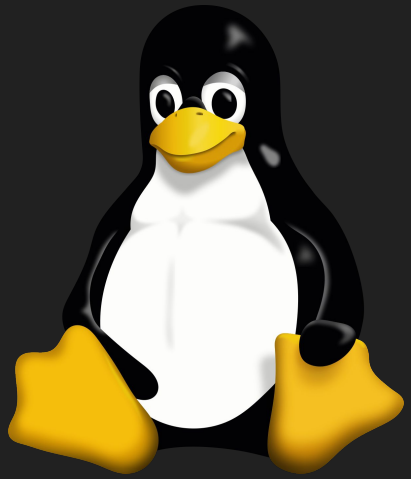
Linux seminar

Infogroep

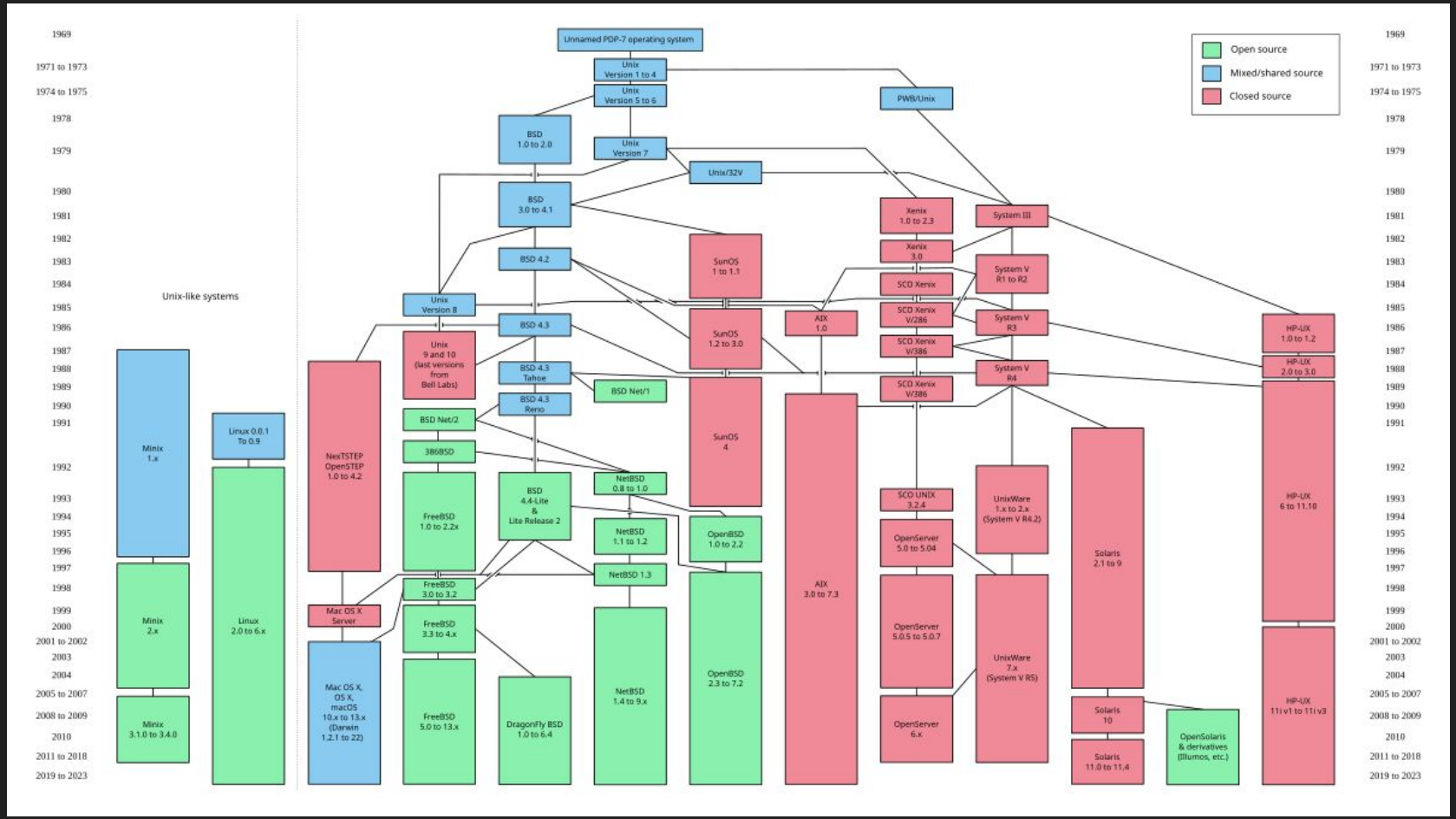
These slides can be found at: https://docs.google.com/presentation/d/1duoq__wPzHDHAFqEBniy49Nk2_jRsmSrl8P1Obo5fao/edit#slide=id.p

Some context

- Made by Linus Torvalds
- Runs on most servers
- UNIX like OS



Linux is not an OS


Open source
Mixed/shared source
Closed source
Unix-like systems
1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2002
2003
2004
2005 to 2007
2008 to 2009
2010
2011 to 2018
2019 to 2023
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
Unix Version 7
PWB/Unix
Unix/32V
BSD 1.0 to 2.0
BSD 3.0 to 4.1
BSD 4.2
BSD 4.3
BSD 4.3 Tahoe
BSD 4.3 Reno
Unix Version 8
Unix 9 and 10 (last versions from Bell Labs)
BSD Net/1
BSD Net/2
386BSD
FreeBSD 1.0 to 2.2x
FreeBSD 3.0 to 3.2
FreeBSD 3.3 to 4.x
FreeBSD 5.0 to 13.x
BSD 4.4-Lite & Lite Release 2
NetBSD 0.8 to 1.0
NetBSD 1.1 to 1.2
NetBSD 1.3
NetBSD 1.4 to 9.x
DragonFly BSD 1.0 to 6.4
OpenBSD 1.0 to 2.2
OpenBSD 2.3 to 7.2
NextSTEP OpenSTEP 1.0 to 4.2
Mac OS X Server
Mac OS X, OS X, macOS 10.x to 13.x (Darwin 1.2.1 to 22)
SunOS 1 to 1.1
SunOS 1.2 to 3.0
SunOS 4
AIX 1.0
AIX 3.0 to 7.3
Xenix 1.0 to 2.3
Xenix 3.0
SCO Xenix
SCO Xenix V/286
SCO Xenix V/386
SCO UNIX 3.2.4
OpenServer 5.0 to 5.04
OpenServer 5.0.5 to 5.0.7
OpenServer 6.x
System III
System V R1 to R2
System V R3
System V R4
UnixWare 1.x to 2.x (System V R4.2)
UnixWare 7.x (System V R5)
Solaris 2.1 to 9
Solaris 10
Solaris 11.0 to 11.4
OpenSolaris & derivatives (Illumos, etc.)
HP-UX 1.0 to 1.2
HP-UX 2.0 to 3.0
HP-UX 6 to 11.10
HP-UX 11 v1 to 11 v3
Minix 1.x
Minix 2.x
Minix 3.1.0 to 3.4.0
Linux 0.0.1 To 0.9
Linux 2.0 to 6.x

Linux distributions

- Debian
  - Ubuntu
  - Linux Mint
- Arch
  - Manjaro
- Red hat
  - Fedora
- …

Package managers

- APT
  - debian
  - ubuntu
  - ...
- pacman
  - arch
  - manjaro
  - ...
- dnf
  - red hat
  - fedora
  - ...

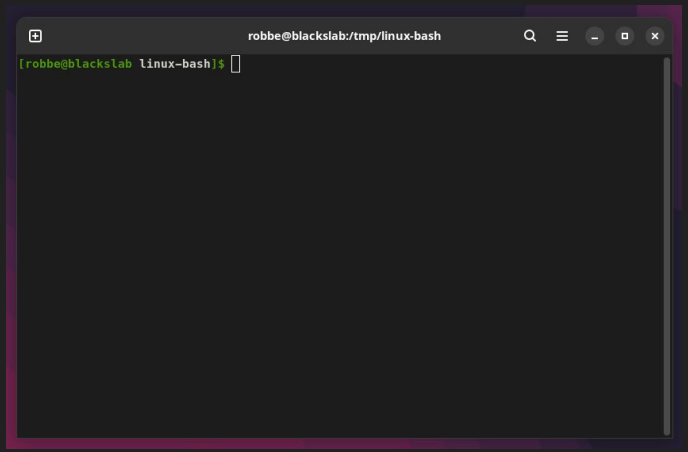
Package managers - installing packages

- Updating local index of packages
  - apt update
- Searching for a package
  - apt search <package-name>
- Installing a package
  - apt install <package-name>
- Removing a package
  - apt remove <package-name>

Examples are for APT

The terminal

- Text based interface with OS
- Usually faster than GUI but more cumbersome
- Also called “a shell”
- Necessary (most of the time) to work on remote machines



Putting it together - package manager

- You heard about a cool program called figlet
- Try to install it on your own using terminal

Solution

apt update
apt search figlet (optional)
apt install figlet

Moving around

Your terminal session has a **current working directory**

Changing it allows you to move around in the system

pwd	**-** Print current working directory
cd <dir>	**-** Change directory
ls	- List directory

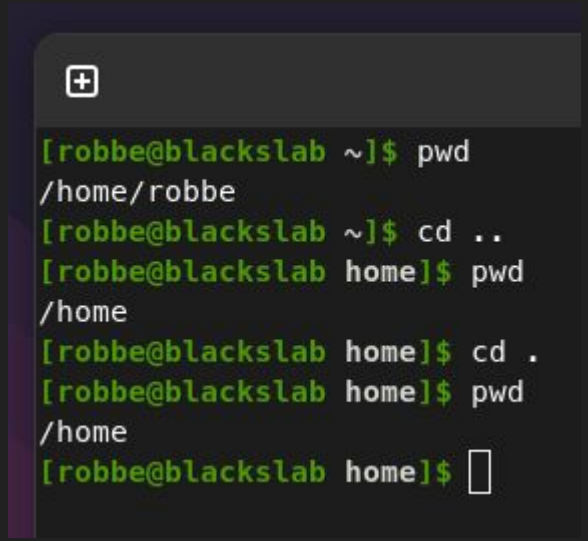
Special directories

Linux has some “special directories”

‘..’ - Refers to the parent directory

‘.’ - Refers to the current directory

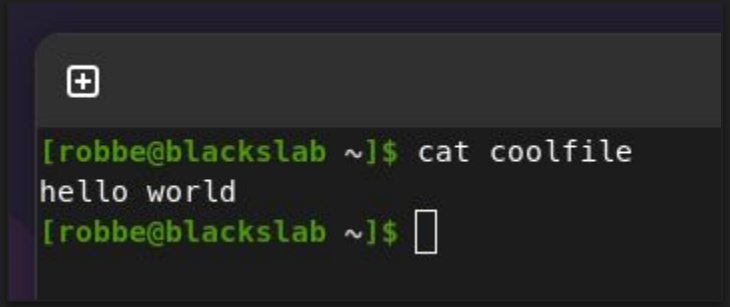
This allows us to move around more easily



Reading files

Reading files in the terminal is relatively simple using cat

cat - Stands for “concatenate files” and prints the given files to terminal



Writing files

Writing files can be done in many ways, the easiest is a text editor!

nano - Nano text editor, usually pre installed on almost any machine

vim - Vi Improved text editor, usually needs to be installed

HELP I opened a file and I cannot get out

In nano: “CTRL+X” (it then asks if you want to save) press Y and enter and you are out

in vim: ESC to go out of edit mode, write “:wq” to write and quit

Connecting with remote servers

SSH - Secure shell

Usually necessary if you want to talk to remote servers

ssh username@server

Putting it all together - remote server

Log into **wilma.vub.ac.be** with your netID and password, and write “hello world” in “test” and afterwards print it to the terminal

Solution

ssh rdegreef@wilma.vub.ac.be
vim test
cat test

Useful commands

- `ls <dir>`
  - List all files in given directory (if the <dir> is omitted, the current directory will be listed)
- `cat <file>*`
  - Print the content of all given files to the screen
- `cd <dir>`
  - Change current working directory to a given directory
- `cd`
  - Change current working directory with your home directory
- `grep <pattern>`
  - Search for a pattern in the given input, for example `cat file.txt | grep hello` will list all references of `hello` in the file.txt. This is because the output of cat will be **piped** into the input of grep.
- `clear`
  - Clears the terminal screen

Useful commands

- `head` `head -n <number>`
  - Print the first N amount of lines of the given input. E.g. `cat test | head -n 5` prints the first 5 lines of the file `test` since it uses the output of cat as input of head
- `tail` `tail -n <number>`
  - Same as head but the last N amount of lines of a given input.
- `history`
  - Prints the entire history of all commands you ever executed (sortof).
    - Use for example `history | tail -n 100` to see the last 100 commands you ran
    - Use for example `history | grep ssh` to see all the commands you ever executed which contained the string “ssh”
- `file <filename>`
  - Gives you information about a file and describes which type of file this is

Useful commands

- `sudo <cmd>`
  - Execute a command with root (administrator for the windows users) privileges
- `man <cmd>`
  - If available, provides a manual page for a given command, for example `man cat` gives you all possible arguments you can provide for the `cat` command
- `less`
  - Buffers input and allows you to scroll up and down in terminal. E.g. `cat test | less` make the output of the `test` file easier to read and scrollable. You can exit with `q`
- `more`
  - Simpler version of `less`, generally using less is advised.

Useful commands

- `kill <pid>`
  - Stop a process by the given process ID
- `ps -a`
  - List all processes on the system
- `pkill <name>`
  - Kill all processes that contain the given <name>
- `echo something`
  - Prints `something` to the screen

Useful tips

- Use TAB in the terminal for autocomplete
- Use the up and down arrows to move through previous executed commands
- If you want to stop a running program use CTRL+C
  - If that doesn’t work try CTRL+D
- A ‘~’ in a path is a shorthand for your home directory.
  - `cat ~/hello` would be equivalent to printing the file hello in your home directory
- `$HOME` also refers to your home directory

Terminal gymnastics

Linux ideologies

- Everything is a file
  - Device drivers
  - Pipes

Extras

- Piping
- Keeping sessions alive using tmux & screen
- Anything else you want to know?
- Permissions
- Other useful commands