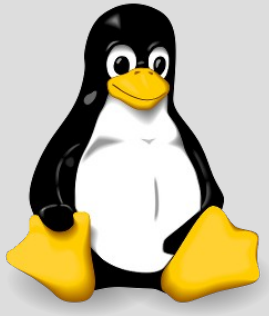
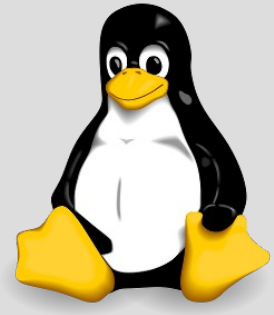


Introduction



J. M. P. Alves

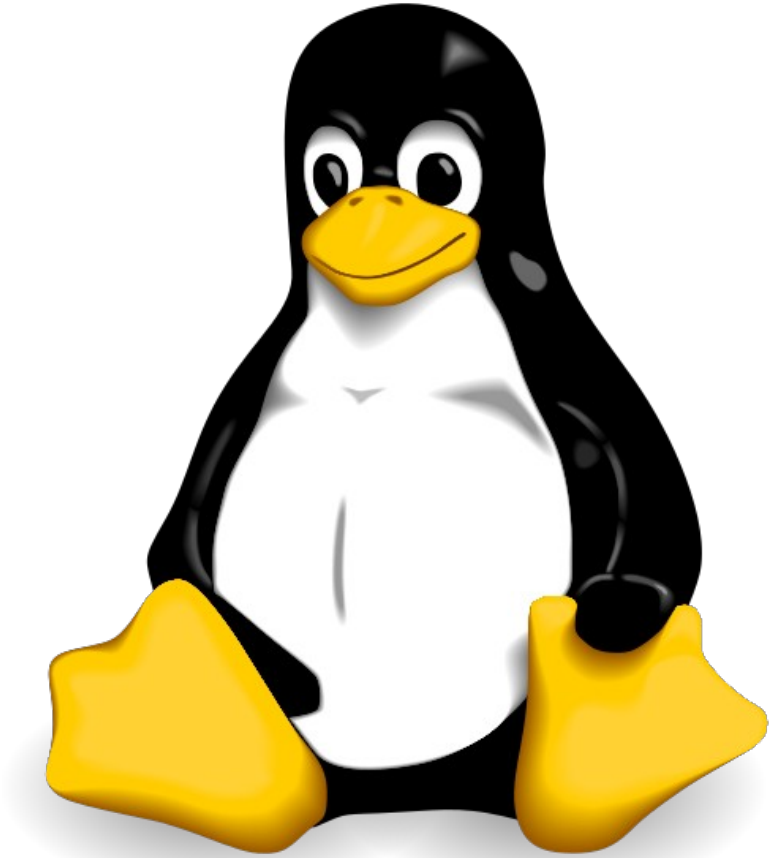
Laboratory of Genomics & Bioinformatics in Parasitology

Department of Parasitology, ICB, USP

- **Introduction to computers and computing (UNIX)**
- **Linux basics**
- **Introduction to the Bash shell**
- **Connecting to this course's virtual machine**

Tux

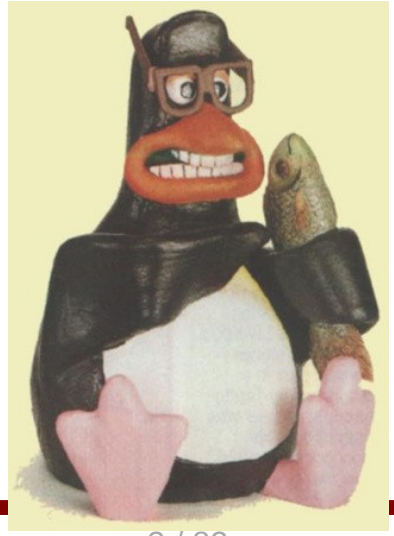
The Linux mascot



By Larry Ewing, 1996

"TUXedo"...

...or Torvalds UniX



Tux's ancestor

Linux (Unix) & science

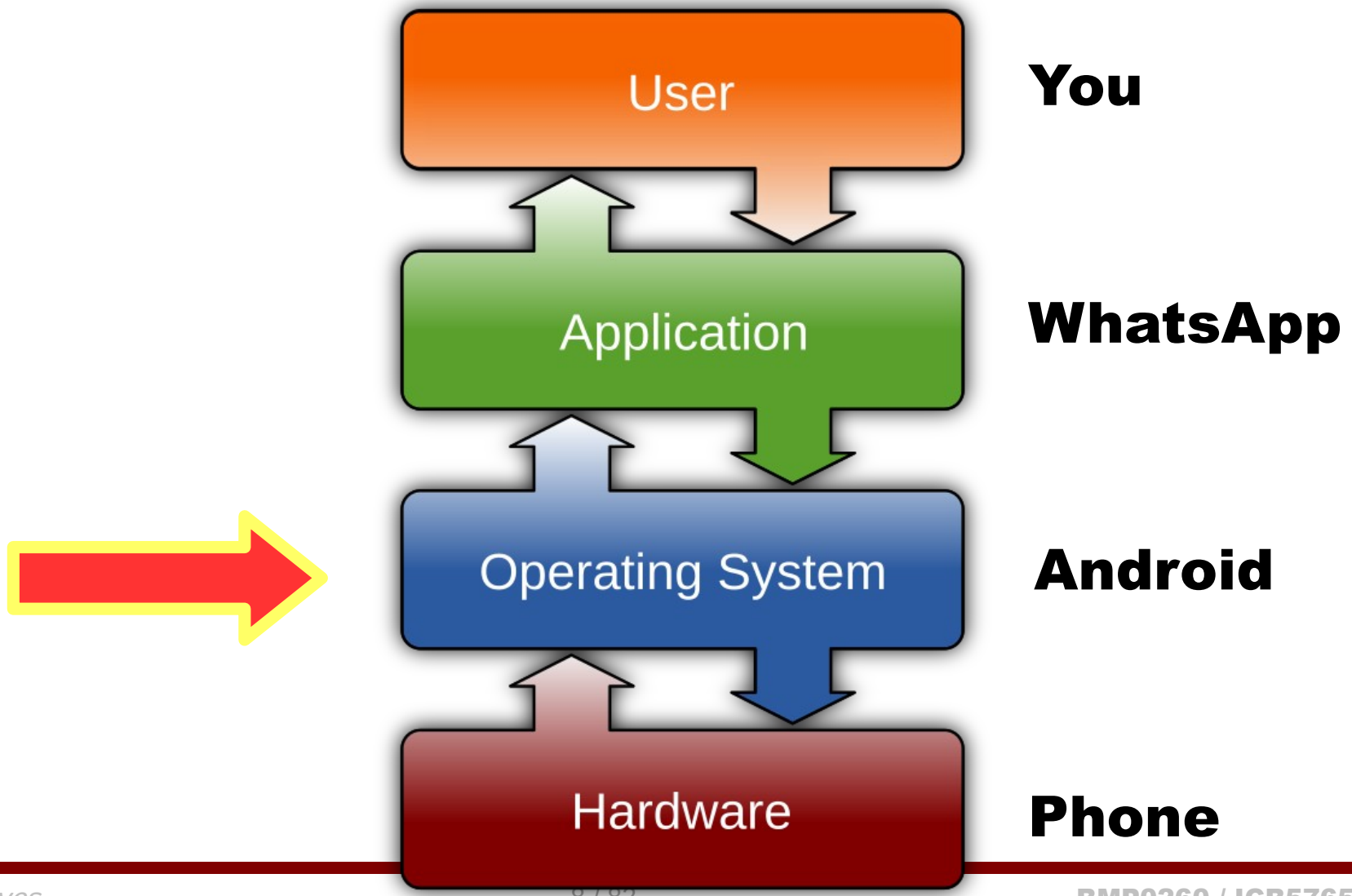
Why so popular together?

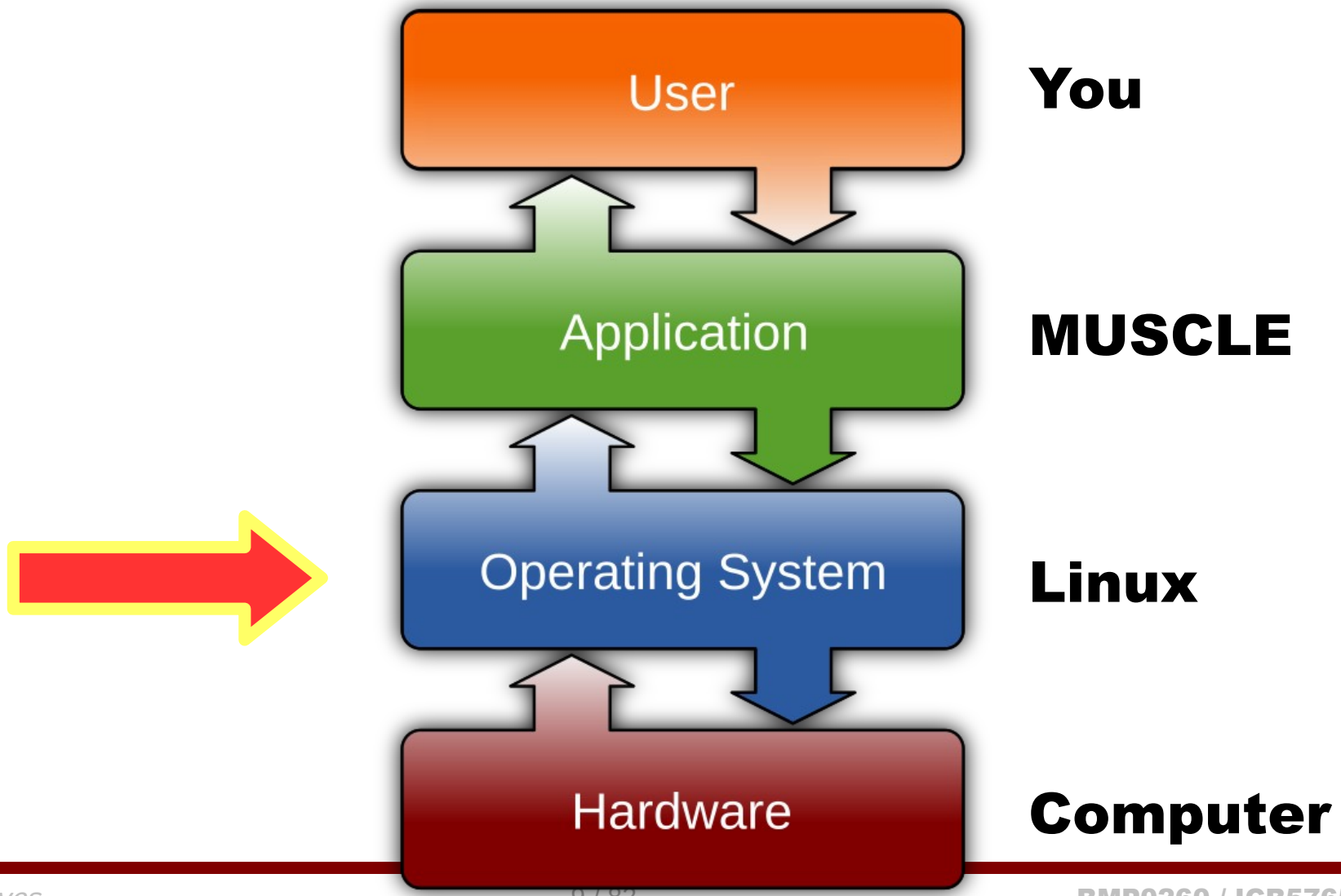
- Historical reasons (programs made for Unix/Linux)
- Available on any kind of computer, especially powerful servers
- Works efficiently with humongous text files (`head`, `tail`, `sort`, `cut`, `paste`, `grep`, etc.)
- Complicated tasks can be made easy by concatenating simpler commands (piping)
- Creating new programs is easy – tools just one or two commands (or clicks) away (`gcc`, `g++`, `python`, `perl`)
- Stable, efficient, open (free software), no cost (software for free)

What IS this Linux, anyway?

Operating system

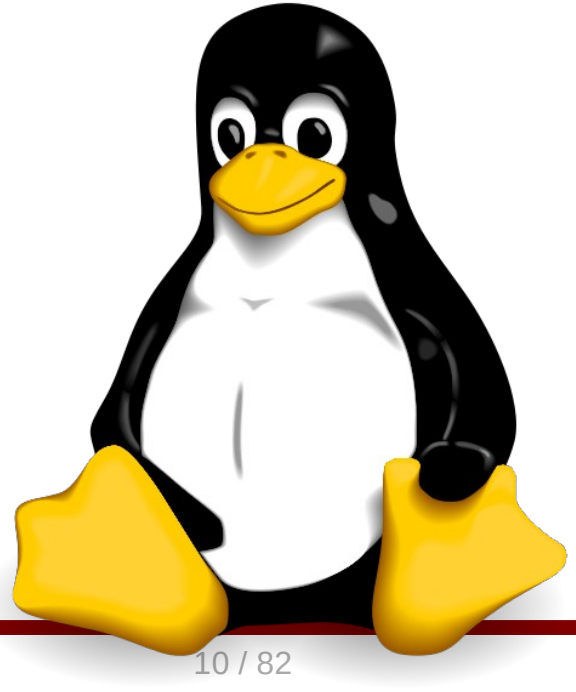
An operating system is a collection of programs that initialize the computer's hardware, providing basic instructions for the control of devices, managing and scheduling tasks, and regulating their interactions with each other.

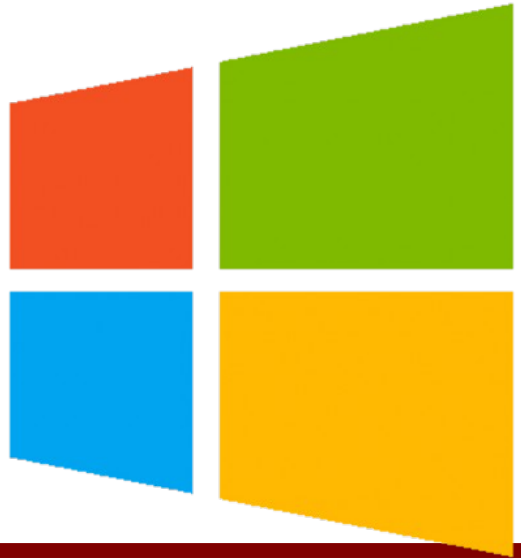

User
Application
Operating System
Hardware
You
WhatsApp
Android
Phone


User
Application
Operating System
Hardware
You
MUSCLE
Linux
Computer

macOS

formerly:





History

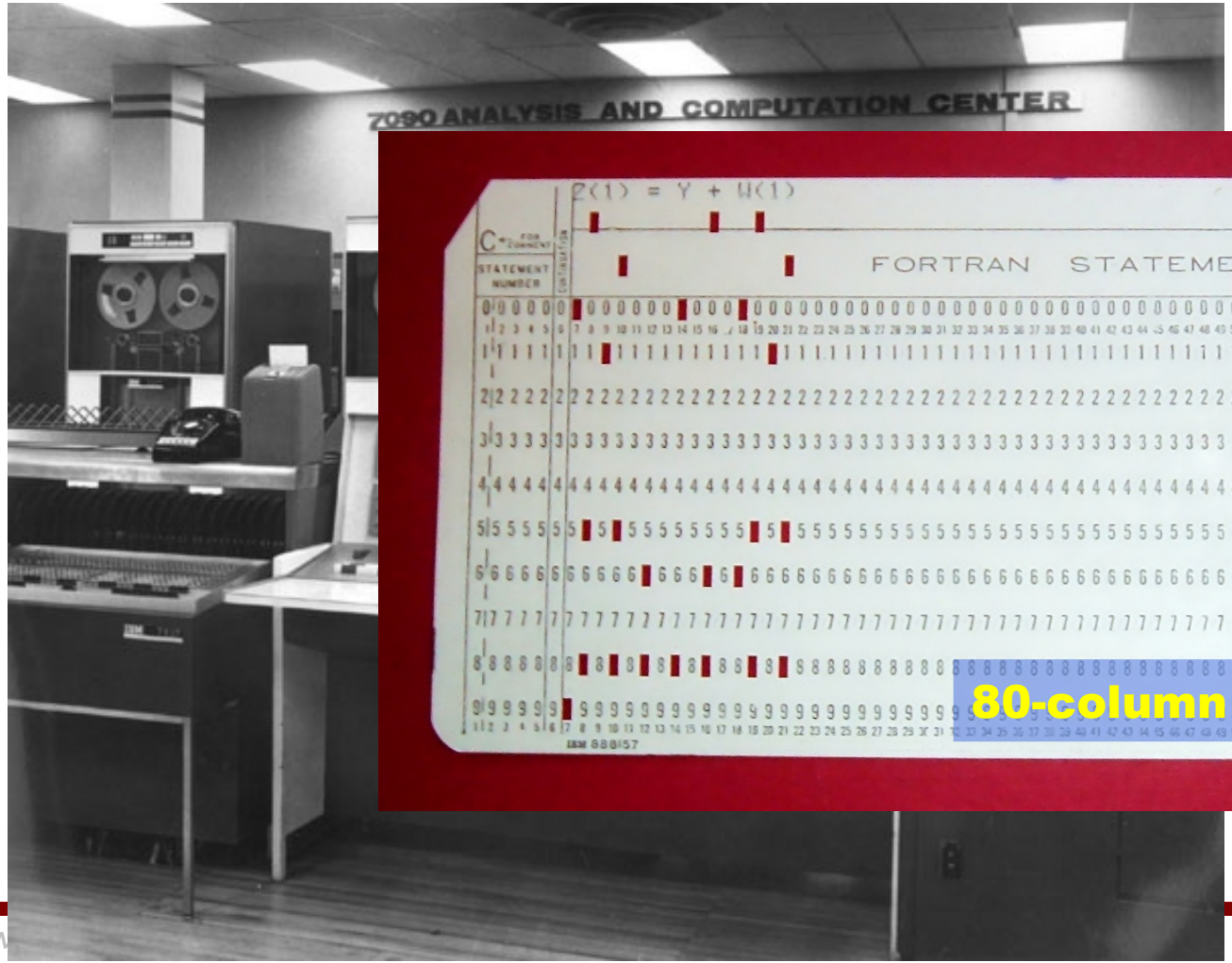
CTSS – *compatible time-sharing system*

Fernando J. Corbató, Marjorie M. Daggett, Robert C. Daley

- Developed at the MIT (Massachusetts Institute of Technology, USA)
- Introduced in November 1961, running on an IBM 7090
- In 1963, already worked very well and was used by scientific researchers for heavy computing
- Several people can use the machine at the same time (time-sharing)

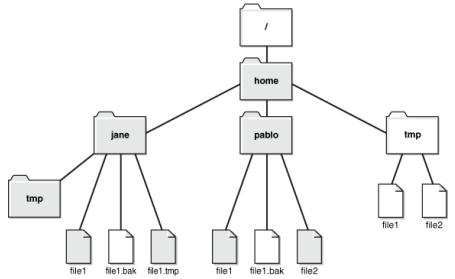
CTSS – *compatible time-sharing system*

Fernando J. Corbató, Marjorie M. Daggett, Robert C. Daley

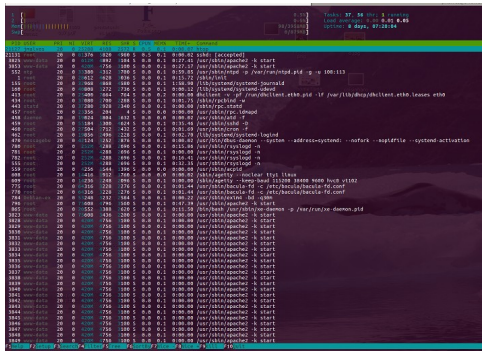


MULTICS – multiplexed information and computing service

- Developed at MIT (Fernando J. Corbató), General Electric and AT&T Bell Labs, from 1964 to 1969 (last system ran until 2000)
- Introduced several innovations that were influential in the development of all other operating systems
  - Hierarchical file system (“tree”)
  - Shell – “covers” the kernel; layer of user-computer interaction



- Tried to increase the system's versatility and flexibility
- Very complex and heavy on computational resources



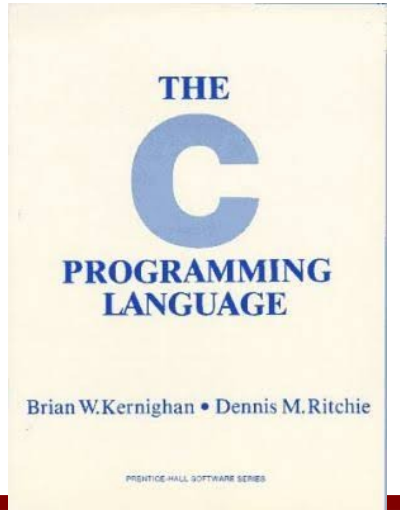
UNICS – uniplexed information and computing service

UNIX

- Bothered by Multics' complexity and inefficiency, Bell Labs researchers slowly abandoned the development consortium

- In 1969, (former Multics) programmers Ken Thompson, Dennis Ritchie, M. D. McIlroy and J. F. Ossanna created a new, simpler system, but with similar goals

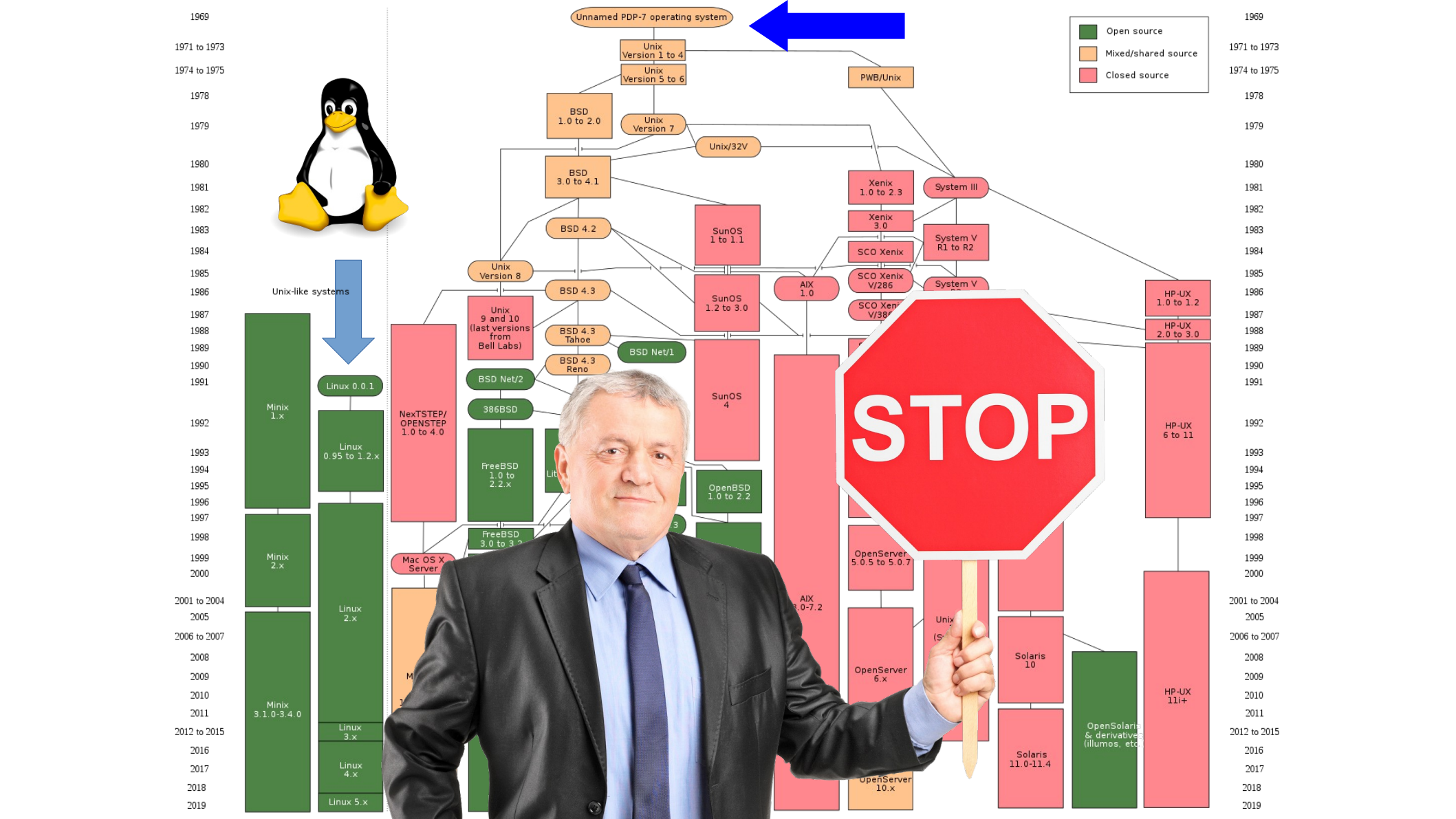
- Originally, it was supposed to be just a tool for programmers to use in their development of applications for other systems



UNIX

- However, the system grew in usage in academia (code freely shared), with users creating their own tools and redistributing the system
- The Unix philosophy: combining small programs that each do only one thing (but do it well), instead of having large programs that do a lot of things (but not as well done):

> *"the power of a system comes more from the relationships among programs than from the programs themselves"*
>
> – Brian W. Kernighan & Rob Pike, 1984


Open source
Mixed/shared source
Closed source
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix Version 7
Unix/32V
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
BSD 4.2
Xenix 3.0
SunOS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix Version 8
BSD 4.3
SunOS 1.2 to 3.0
AIX 1.0
SCO Xenix V/286
System V
HP-UX 1.0 to 1.2
Unix 9 and 10 (last versions from Bell Labs)
BSD 4.3 Tahoe
BSD Net/1
HP-UX 2.0 to 3.0
BSD 4.3 Reno
BSD Net/2
SunOS 4
386BSD
HP-UX 6 to 11
FreeBSD 1.0 to 2.2.x
OpenBSD 1.0 to 2.2
FreeBSD 3.0 to 3.2
Mac OS X Server
OpenServer 5.0.5 to 5.0.7
AIX 3.0-7.2
OpenServer 6.x
Solaris 10
HP-UX 11i+
Solaris 11.0-11.4
OpenSolaris & derivatives (illumos, etc)
OpenServer 10.x
Unix-like systems
Minix 1.x
Minix 2.x
Minix 3.1.0-3.4.0
Linux 0.0.1
Linux 0.95 to 1.2.x
Linux 2.x
Linux 3.x
Linux 4.x
Linux 5.x
NexTSTEP/ OPENSTEP 1.0 to 4.0
STOP
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 2004
2005
2006 to 2007
2008
2009
2010
2011
2012 to 2015
2016
2017
2018
2019

Not shown in the figure above, but just as important

GNU

http://www.gnu.org/

September 1983 – Richard M. Stallman

From CSvax:pur-ee:inuxc!ixn5c!ihnp4!houxm!mhuxi!eagle!mit-vax!mit-eddie!RMS@MIT-OZ
From: RMS%MIT-OZ@mit-eddie
Newsgroups: net.unix-wizards,net.usoft
Subject: new Unix implementation
Date: Tue, 27-Sep-83 12:35:59 EST
Organization: MIT AI Lab, Cambridge, MA

Free Unix!

Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for **Gnu's Not Unix**), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed.

To begin with, **GNU will be a kernel plus all the utilities** needed to write and run C programs: editor, shell, C compiler, linker, assembler, and a few other things. After this we will add a text … … …

https://www.gnu.org/gnu/initial-announcement.html

GNU GPL

https://en.wikipedia.org/wiki/GNU_General_Public_License

Free (?) software

Free beer

X

GPL Free Software

Free as in Freedom

Free speech

Free Software

It means the user has freedoms (nothing to do with price)

The four essential freedoms (GPL)

0) To run the program in any way, anywhere you want

1) To study and modify the program's source code

2) To redistribute original copies of the program

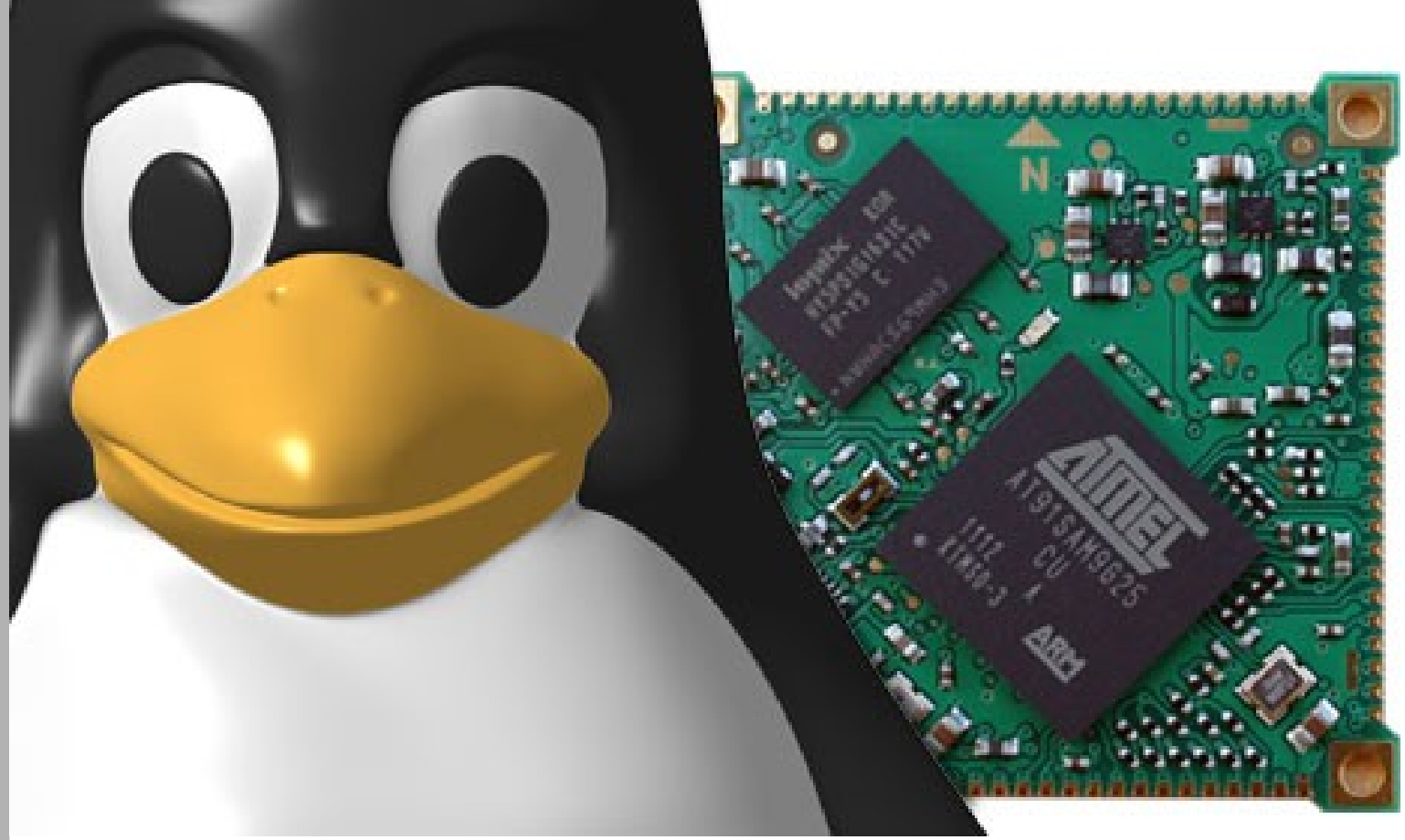
3) To distribute modified copies of the program

https://www.gnu.org/philosophy/philosophy.html

The kernel is the part of operating system that is closest to the hardware

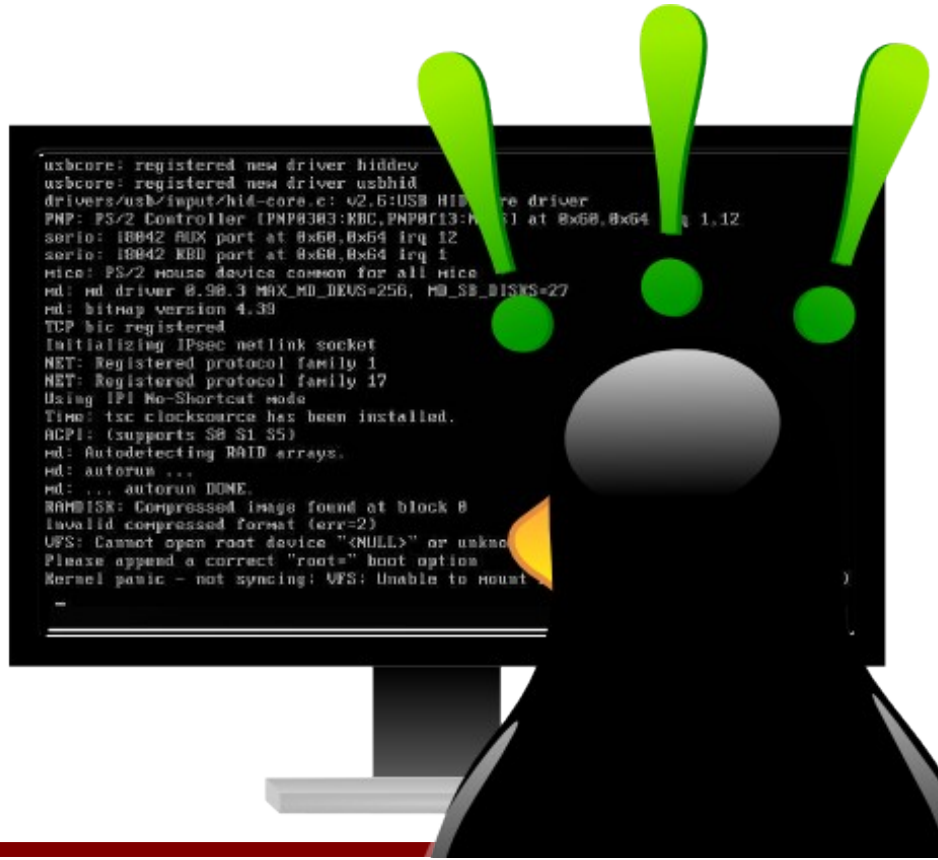
Services provided by the kernel:

- Device drivers (keyboard, screen, printer, disks etc.)
- Process (“program”) manager
- Memory management
- System calls



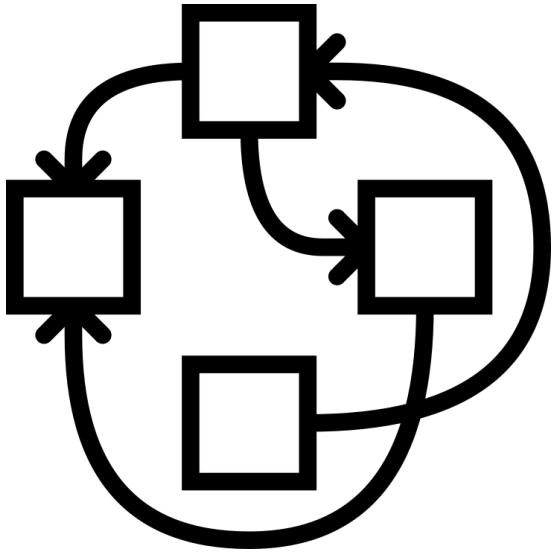
The operating system kernel is the last stronghold between the user and the hardware itself

Kernel panic



GNU's kernel

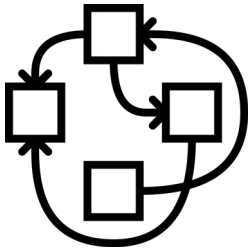
HURD

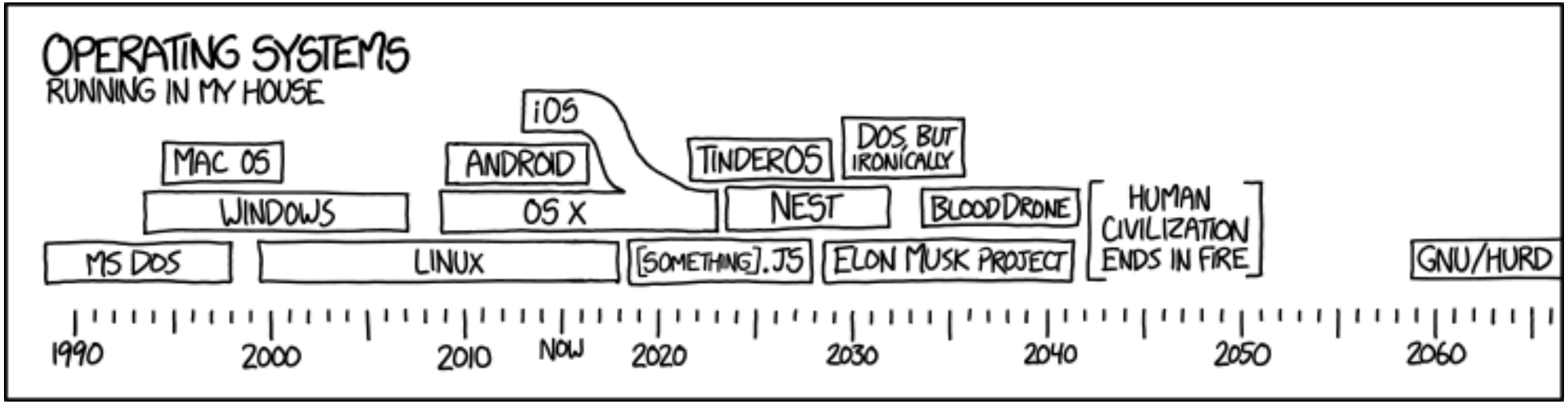


(pun on “herd”)

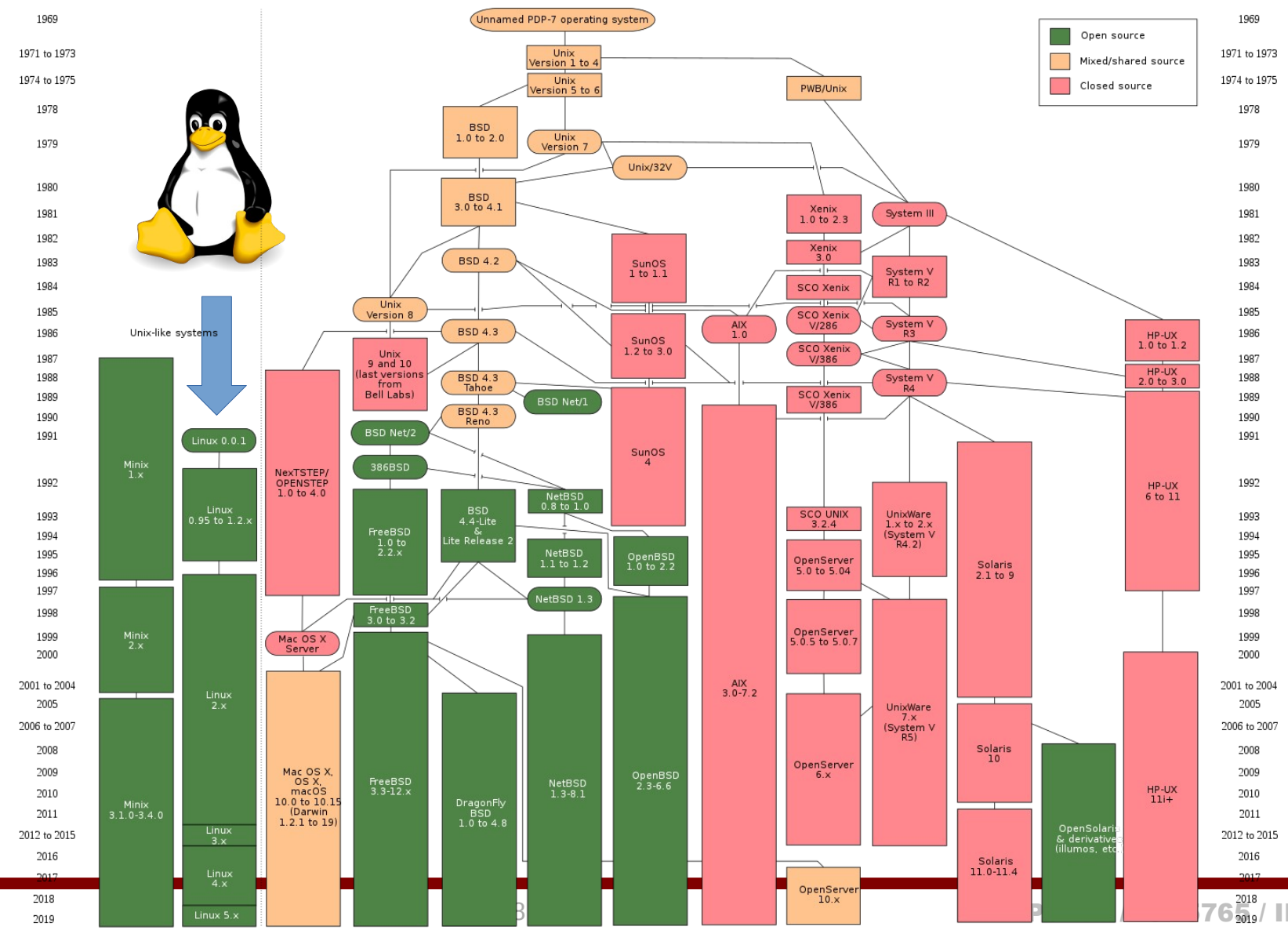
Started in 1990...

...and still unfinished





https://xkcd.com/1508/


Open source
Mixed/shared source
Closed source
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 2.2.x
FreeBSD 3.0 to 3.2
FreeBSD 3.3-12.x
BSD 4.4-Lite & Lite Release 2
NetBSD 0.8 to 1.0
NetBSD 1.1 to 1.2
NetBSD 1.3
NetBSD 1.3-8.1
OpenBSD 1.0 to 2.2
OpenBSD 2.3-6.6
DragonFly BSD 1.0 to 4.8
SunOS 1 to 1.1
SunOS 1.2 to 3.0
SunOS 4
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
OpenServer 10.x
AIX 1.0
AIX 3.0-7.2
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 11.0-11.4
OpenSolaris & derivatives (illumos, etc)
HP-UX 1.0 to 1.2
HP-UX 2.0 to 3.0
HP-UX 6 to 11
HP-UX 11i+
Unix-like systems
Minix 1.x
Minix 2.x
Minix 3.1.0-3.4.0
Linux 0.0.1
Linux 0.95 to 1.2.x
Linux 2.x
Linux 3.x
Linux 4.x
Linux 5.x
NeXTSTEP/ OPENSTEP 1.0 to 4.0
Mac OS X Server
Mac OS X, OS X, macOS 10.0 to 10.15 (Darwin 1.2.1 to 19)
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 2004
2005
2006 to 2007
2008
2009
2010
2011
2012 to 2015
2016
2017
2018
2019

Linus Torvalds

21 year old Finnish student, 1991

Message to the Usenet comp.os.minix newsgroup:

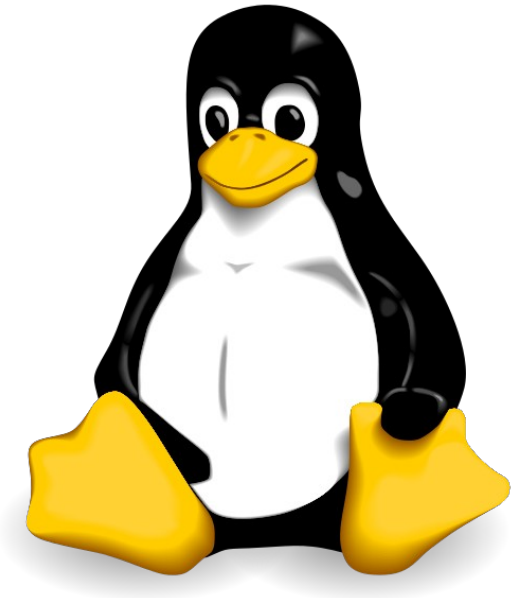
```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
  Newsgroups: comp.os.minix
  Subject: What would you like to see most in minix?
  Summary: small poll for my new operating system
  Message-ID:
  Date: 25 Aug 91 20:57:08 GMT
  Organization: University of Helsinki


   Hello everybody out there using minix -

  I'm doing a (free) operating system (just a hobby, won't be big and
  professional like gnu) for 386(486) AT clones.  This has been brewing
  since april, and is starting to get ready.  I'd like any feedback on
  things people like/dislike in minix, as my OS resembles it somewhat
  (same physical layout of the file-system (due to practical reasons)
  among other things).

  I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
  This implies that I'll get something practical within a few months, and
  I'd like to know what features most people would want.  Any suggestions
  are welcome, but I won't promise I'll implement them :-)

                Linus (torvalds@kruuna.helsinki.fi)

  PS.  Yes - it's free of any minix code, and it has a multi-threaded fs.
  It is NOT protable (uses 386 task switching etc), and it probably never
  will support anything other than AT-harddisks, as that's all I have :-(.
```

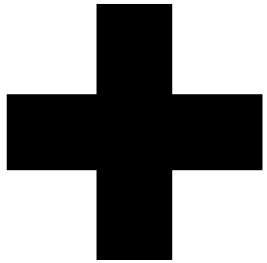
Soon, a community formed around that little project...

- 1993 – First Linux distribution: Slackware
- 1993 – Debian project starts
- 1994 – RedHat and SuSE distributions are released
- 2004 – Ubuntu, based on Debian.
  One of the most popular Linux distributions. Simple to install and use.

DistroWatch

http://distrowatch.com





= GNU / Linux

https://en.wikipedia.org/wiki/GNU/Linux_naming_controversy

Everyone here uses Linux

(...or some other kind of Unix, daily, and not just in this course!)


~86%
SAMSUNG
android

GNU GPL

(and other free software licenses)


SAMSUNG
android


Cockspur Street
Strand
The Mall
3 min
A3212 Whitehall
15
yards
13 M
25 min
11:01
11:26
30 mph
A308 Northumberland Avenue
TOMTOM


Elite


ubuntu
linux for human beings
XPS


tv

Your Facebook photos are here...

http://www.google.com/about/datacenters/gallery/

YouTube videos, Gmail e-mails,
your Google search... come from here

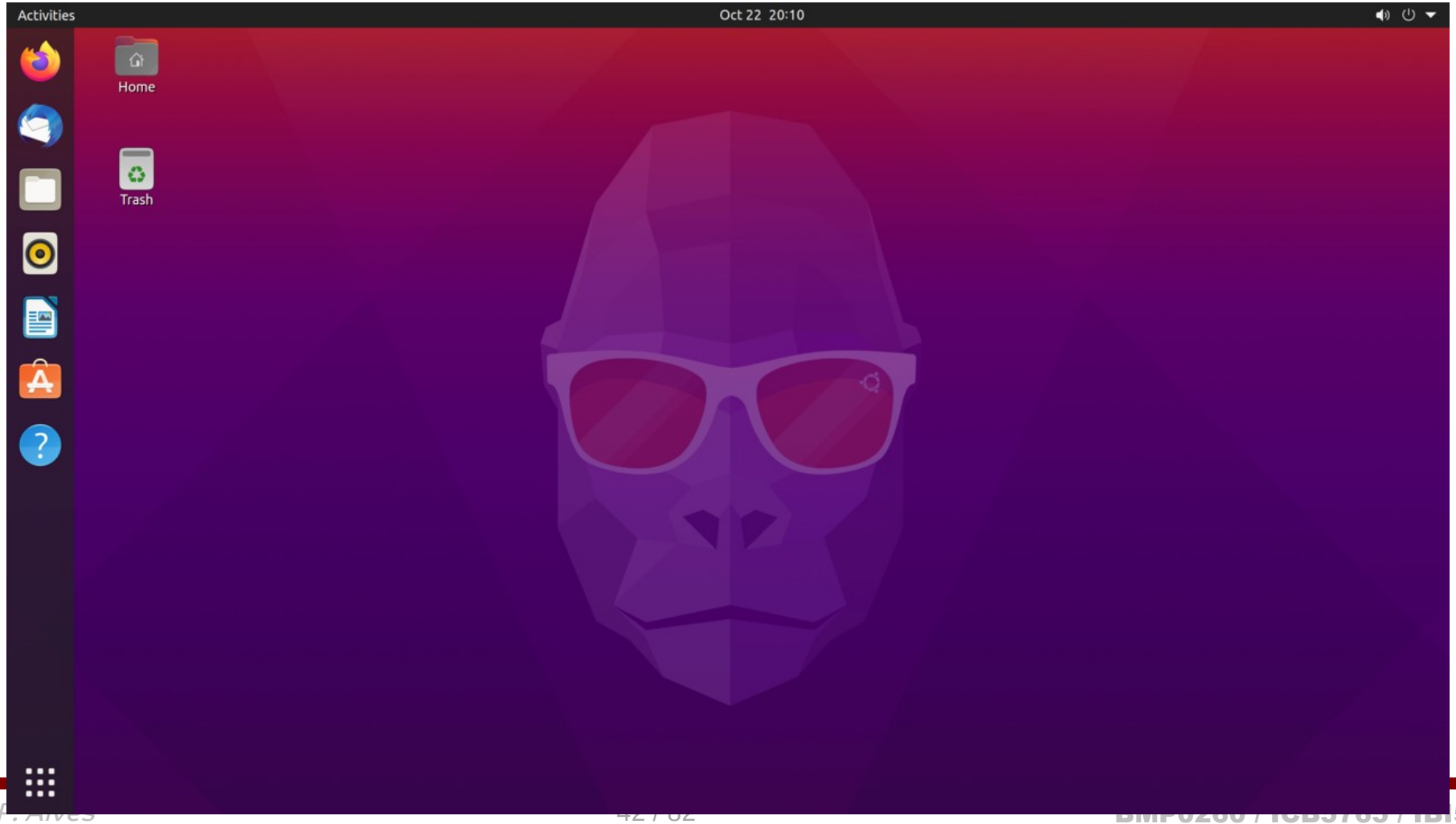
© Google

Linux in action

Unity (old Ubuntu)

https://www.ubuntu.com

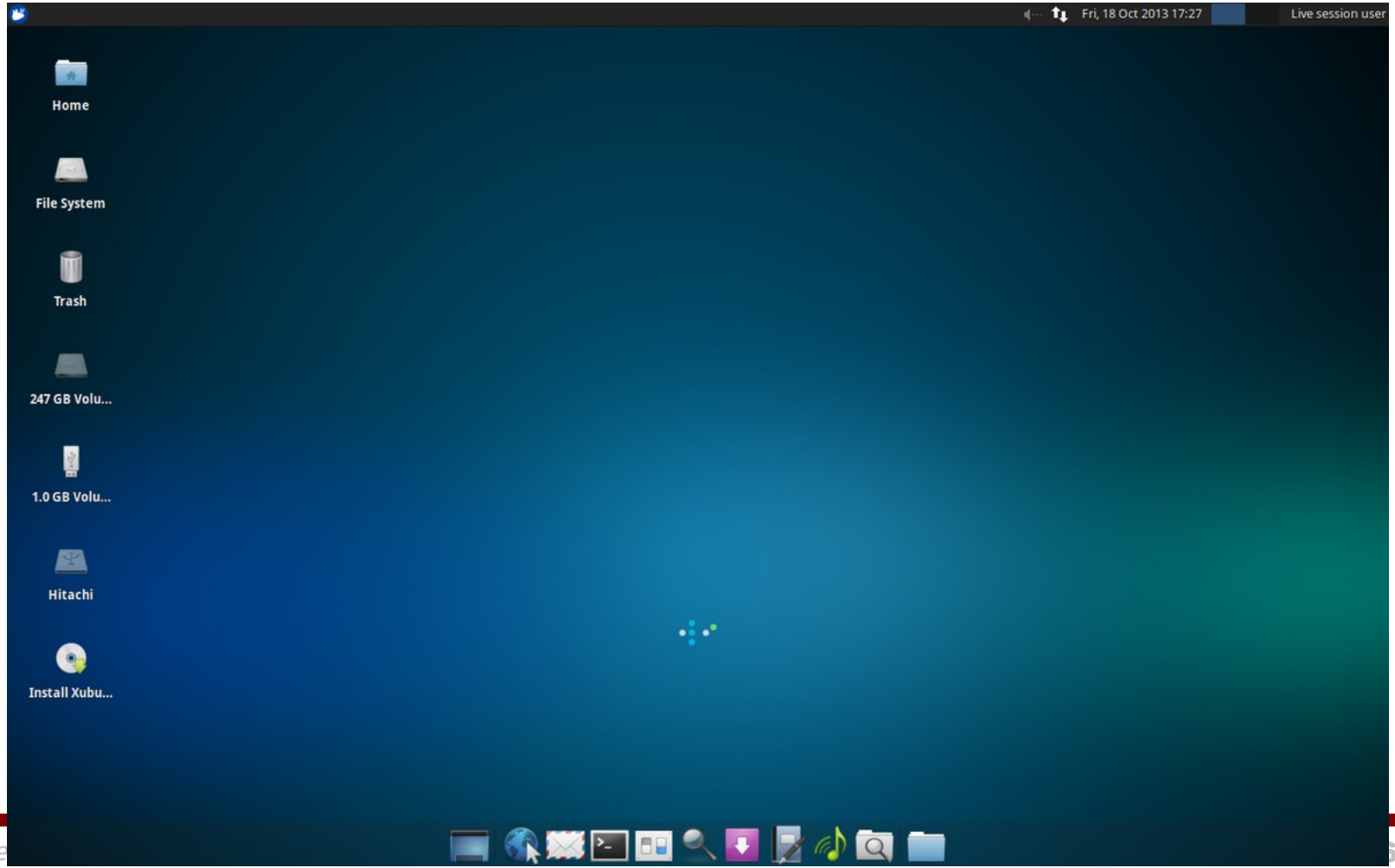
GNOME 3.x (Ubuntu)



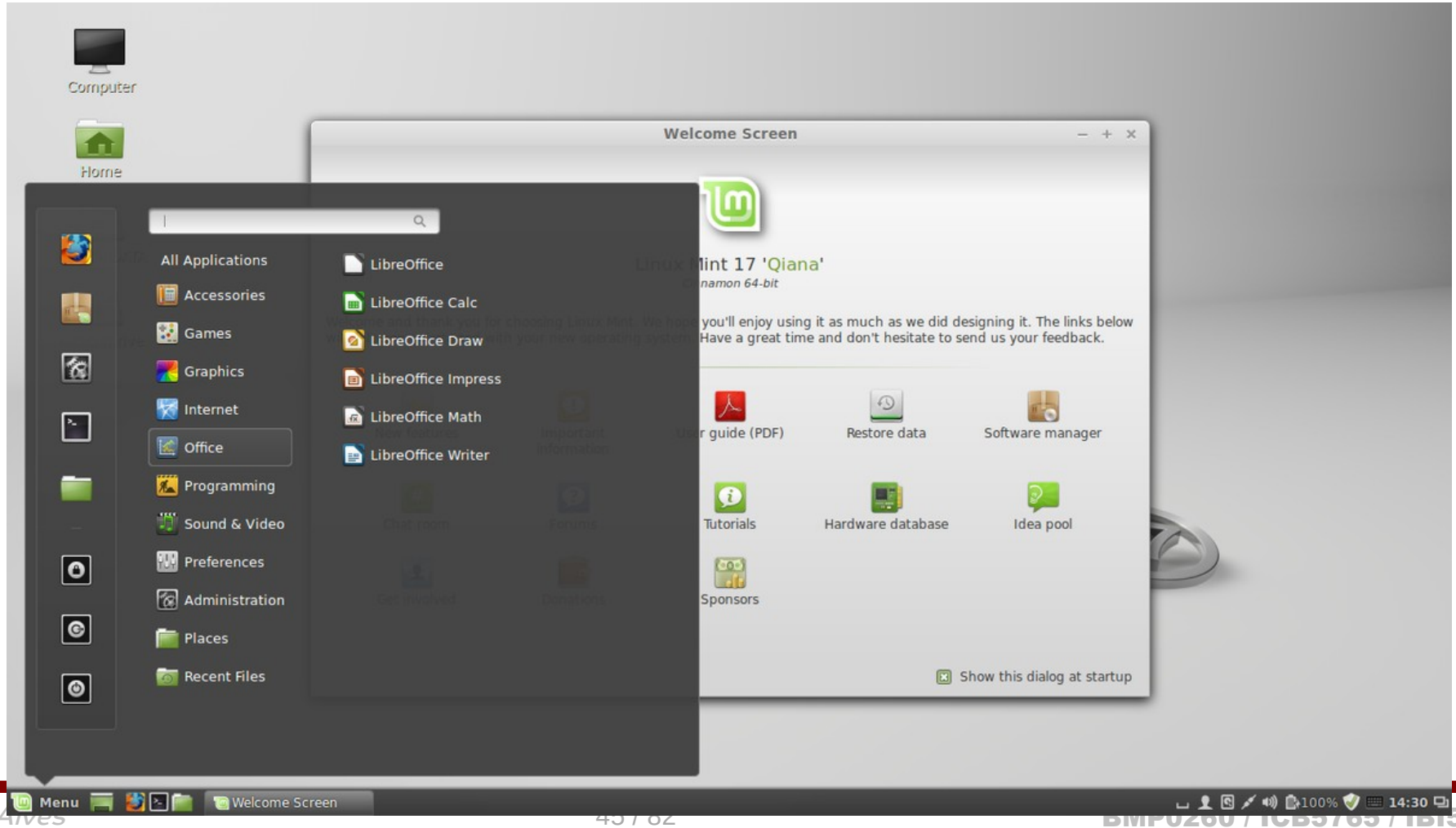
GNOME 3.16


Activities
Thu 13:08
Type to search...
X Window System - Wikipedia, the free encyclopedia
Hong Kong International Airport
33 °C
Scattered clouds
Home

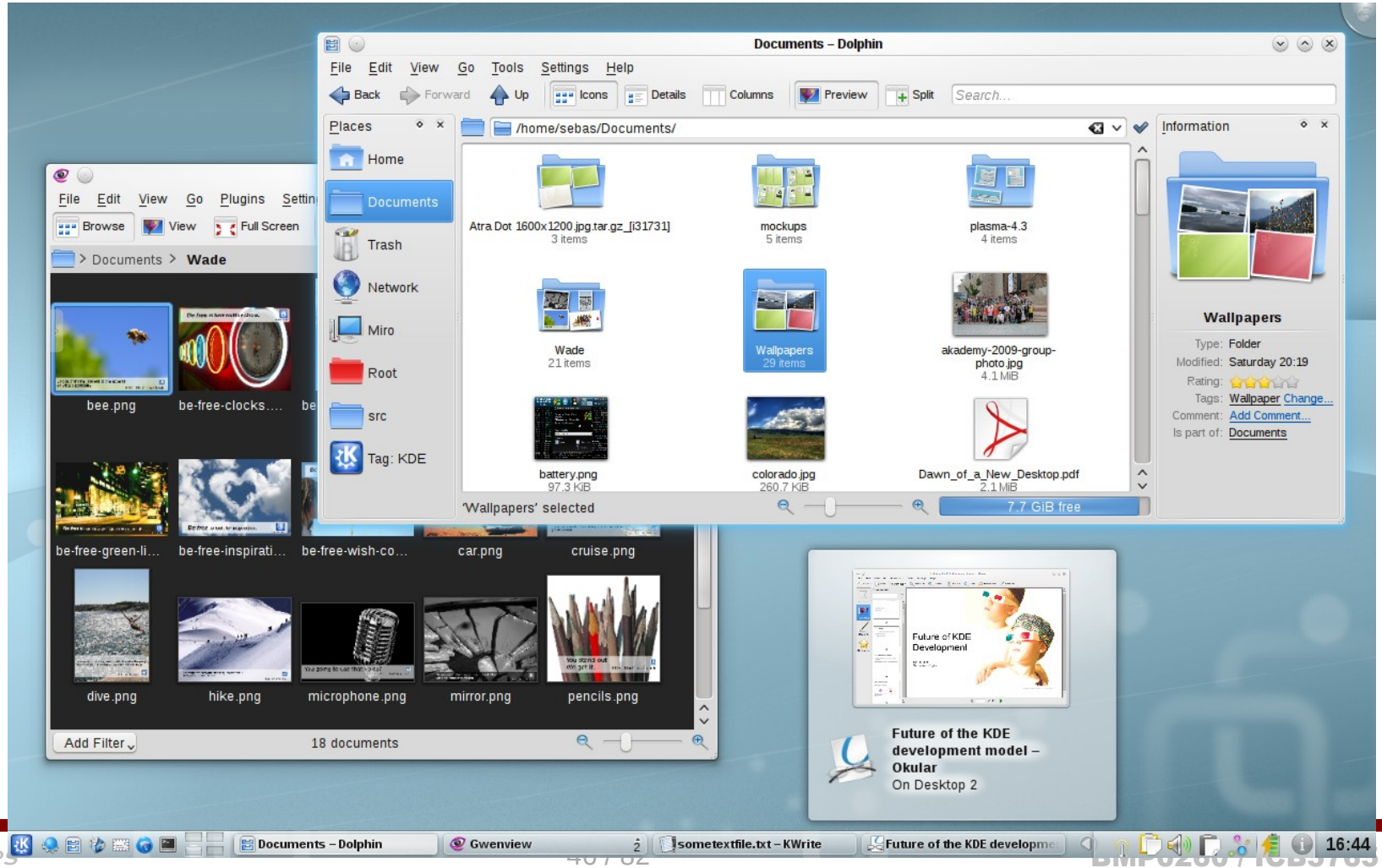
Xfce (Xubuntu)



Cinnamon (Mint)

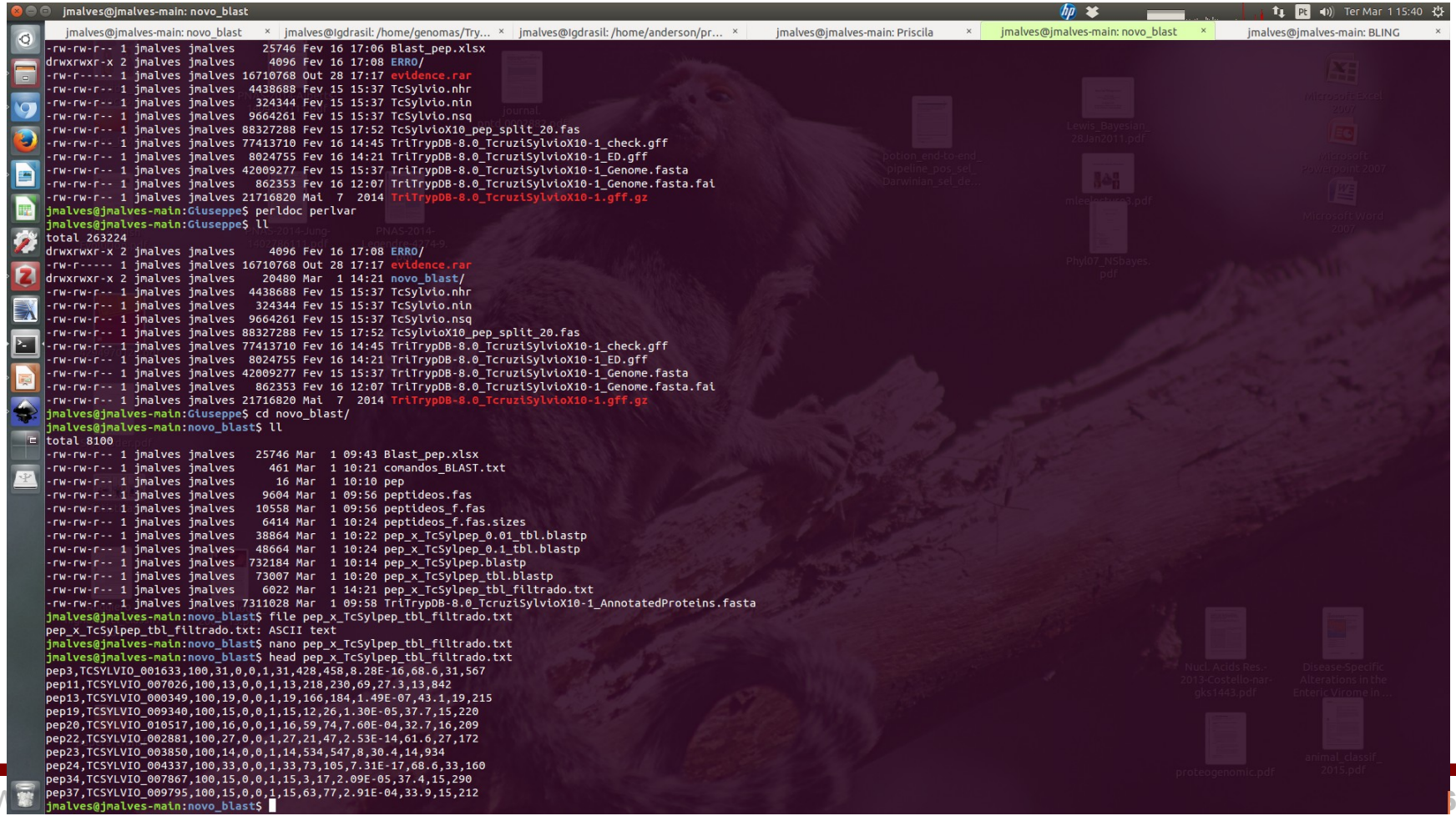


KDE


Documents – Dolphin
File Edit View Go Tools Settings Help
Back Forward Up Icons Details Columns Preview Split Search...
Places
Home
Documents
Trash
Network
Miro
Root
src
Tag: KDE
/home/sebas/Documents/
Atra Dot 1600x1200.jpg.tar.gz_[i31731]
3 items
mockups
5 items
plasma-4.3
4 items
Wade
21 items
Wallpapers
29 items
akademy-2009-group-photo.jpg
4.1 MiB
battery.png
97.3 KiB
colorado.jpg
260.7 KiB
Dawn_of_a_New_Desktop.pdf
2.1 MiB
'Wallpapers' selected
7.7 GiB free
Information
Wallpapers
Type: Folder
Modified: Saturday 20:19
Rating:
Tags: Wallpaper Change...
Comment: Add Comment...
Is part of: Documents
File Edit View Go Plugins Settin
Browse View Full Screen
Documents > Wade
bee.png
be-free-clocks....
be-free-green-li...
be-free-inspirati...
be-free-wish-co...
car.png
cruise.png
dive.png
hike.png
microphone.png
mirror.png
pencils.png
Add Filter
18 documents
Future of KDE Development
Future of the KDE development model – Okular
On Desktop 2
Documents – Dolphin
Gwenview
sometextfile.txt – KWrite
Future of the KDE developme
16:44

Command-line interface

one interface to rule them all, and in the darkness bind them



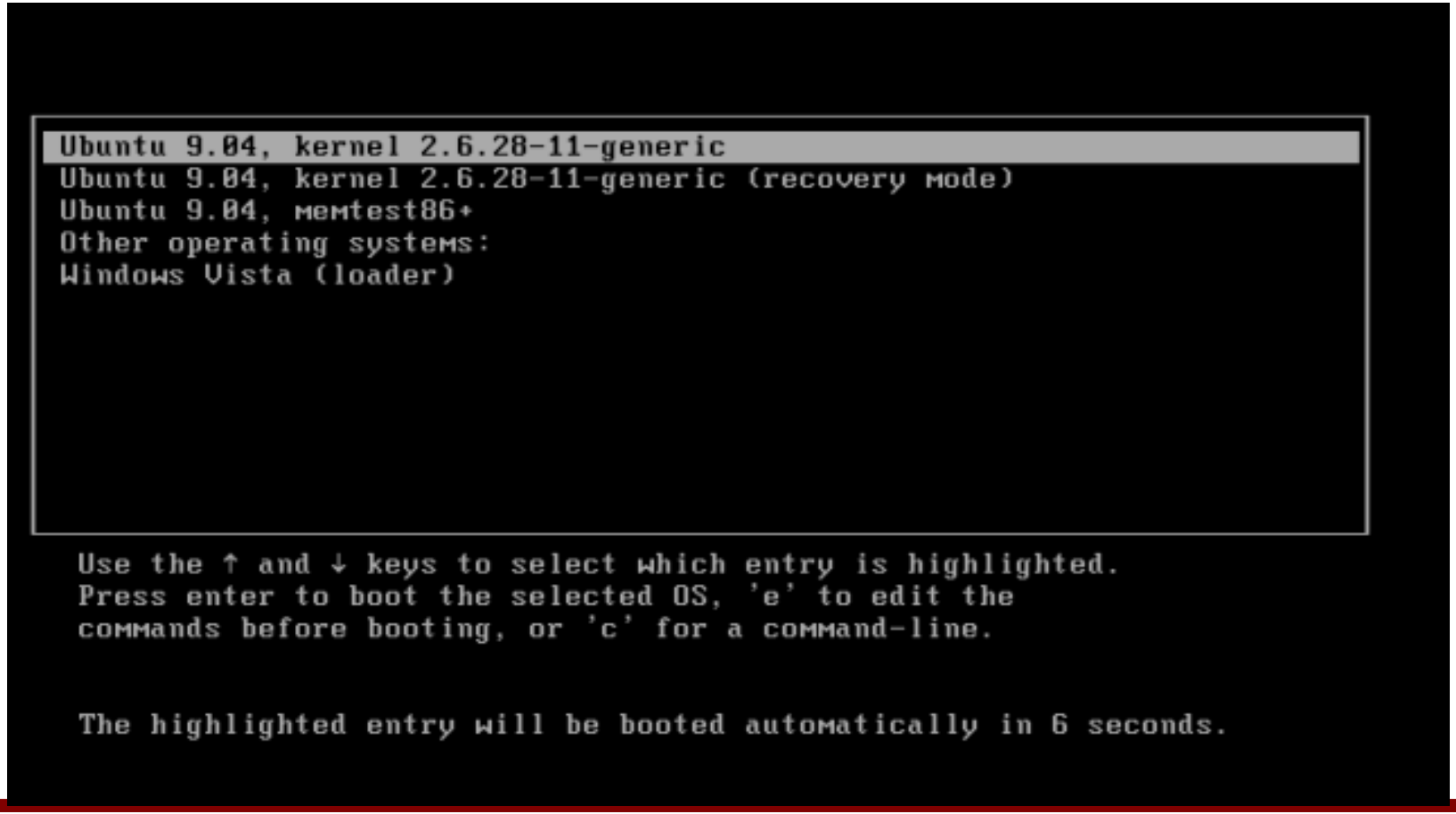
Living together in harmony

or: how to have more than one system in the same machine

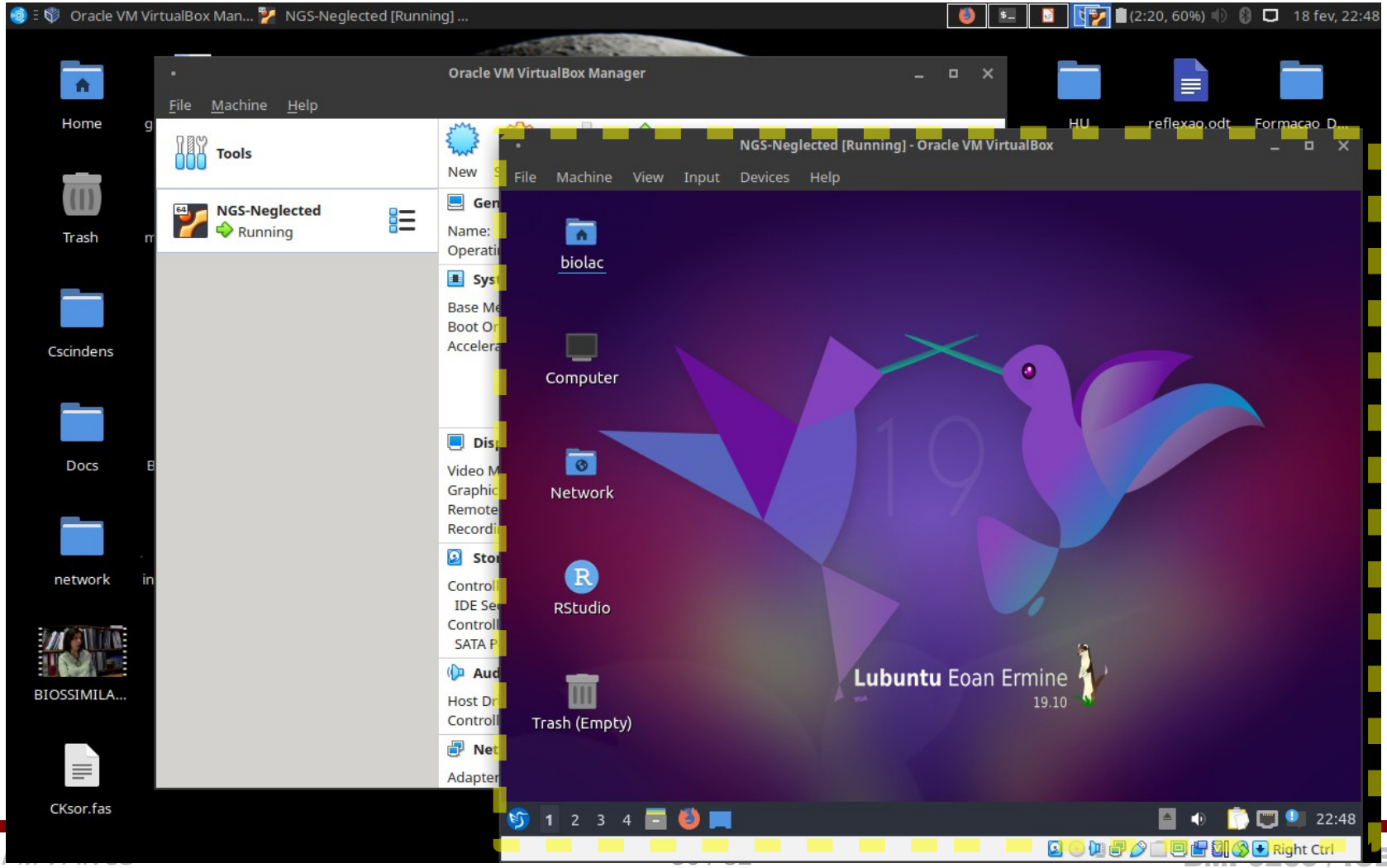
- Dual booting – Windows (using *grub* – Linux boot manager) or Mac (Boot Camp)
- Exporting an X-Window (a.k.a. X11 or just X) terminal
  - Terminal + Xquartz on Mac
  - PuTTY + Xming on Windows
  - X-Win32 (commercial) e MobaXterm (freeware) – X11 clients
- Use a program for remote access of the desktop's graphical interface.
  - X2Go – based on NX project.
  - VNC – several implementations of the protocol
  - Microsoft Remote Desktop Protocol (RDP)
  - ARD – Apple Remote Desktop – derived from VNC
- Use virtual machines (virtualization software, e.g., **VirtualBox**)

Dual booting

one of the ways of running more than one system



Virtual machine



The command-line interface or CLI

- User-computer interaction is performed through text commands and messages, not clicks of the mouse on buttons, menus etc.
- Each command is given as a line of text
- Little used nowadays by "normal" computer users – who prefer *graphical shells* (windows, mouse, buttons, menus etc.)
- Preferred by advanced (and/or vision impaired) users

The command-line interface or CLI

- The CLI is much easier to automate
- It uses way less computational power for the interface itself
- The CLI provides a more concise and powerful way of controlling the computer
- But, paraphrasing Peter Parker's uncle, we can say that with great power comes great complexity!

Why use the CLI?

according to the FSF's "Introduction to the Command-Line"

Flexibility

With graphical programs, you sometimes hit a limit; you just can't do what you want. With the command line, you can combine commands to yield a virtually infinite range of new and interesting functions. By combining commands creatively, you can make the command line do exactly what you want; it puts you in control of your computer.

Why use the CLI?

according to the FSF's "Introduction to the Command-Line"

Reliability

Graphical programs are often immature or even unstable. In contrast, most of the tools that the command line offers are highly reliable. One of the reasons for this reliability is their maturity; the oldest command line programs have been around since the late 1970s. This means that these command lines have been tested for over three decades. They also tend to work the same way across different operating systems.

Why use the CLI?

according to the FSF's "Introduction to the Command-Line"

Speed

Graphics consume a lot of your hardware's resources, often resulting in slowness or instability. The command line, on the other hand, uses the computer's resources much more sparingly, leaving memory and processing power for the tasks that you actually want to accomplish. The command line is also intrinsically faster; instead of clicking through long chains of graphical menus, you can type commands in a dozen or so keystrokes, and often apply them to multiple files or other objects.

Why use the CLI?

according to the FSF's "Introduction to the Command-Line"

Automation

You can also store commands in text files, and use logical controls like in programming. These text files are called scripts and can be used instead of typing a long series of commands each time. For example, if you store commands in a file called `mycommand.sh`, you don't have to type out the commands again. Instead, you can simply type:

```
mycommand.sh
```

Why use the CLI?

according to the FSF's "Introduction to the Command-Line"

Experience

Using the command line is a great learning experience. When you use the command line, you communicate with your computer more directly than with the graphical programs, thus learning a lot about its inner workings.

- The CLI descends from the **teletypes (tty)**, electro-mechanical machines used to exchange information, one line at a time, between distant people
- The teletype was one of the first ways of communicating with computers
- Descendants of the teletype are still used for communication by people with hearing or speech impediments

Teletypewriter, ou TTY

Bash

- The Bourne Again shell, program written for the GNU project and released in 1989
- Evolved from the Bourne shell (**`sh`**, still available for compatibility)
- Bash is a superset of **`sh`**, so mostly everything that works in **`sh`** should work in **`bash`** (but not the other way around!)
- The standard shell of GNU and Apple's macOS

> *"The UNIX shell program interprets user commands, which are either directly entered by the user, or which can be read from a **file** called the **shell script or shell program**."*
>
> *(Bash Beginners Guide)*

Using the local shell running in your own Linux machine is all nice and good...

...but nothing stops you from exploring other machines!

As long as you have **an account** on that Linux machine, and it **accepts remote connections**, the computer can be located anywhere in the world and you will be able to access it using secure shell (`ssh`)

The virtual machine for this course:

Address: `200.144.244.172`

User name: **your first name** (lower case only!)

Temporary password: **the one sent to you**

Special programs, following a secure protocol, allow for a secure remote connection between your machine and the remote shell

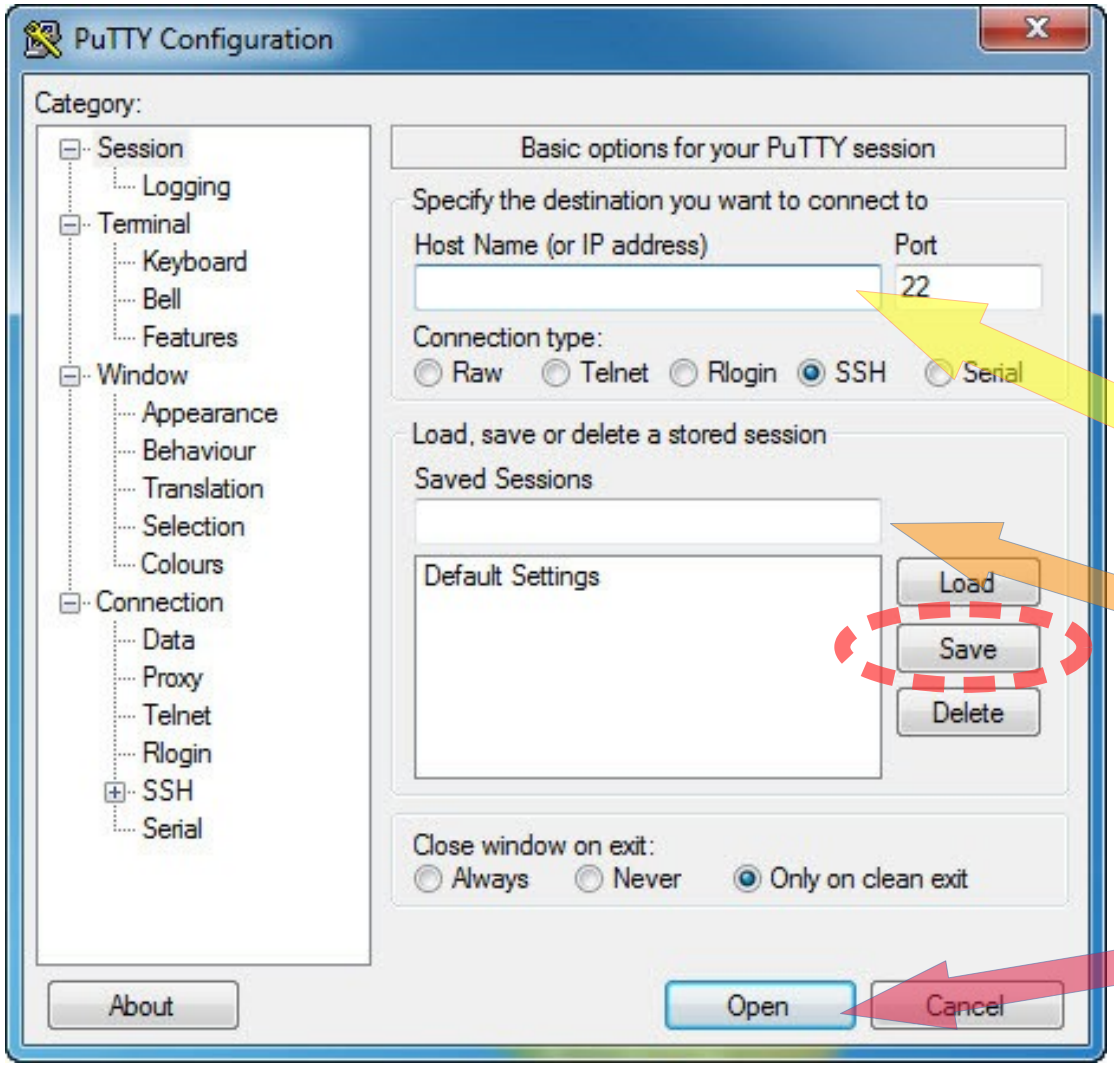
Secure Shell (SSH) is a cryptographic network protocol for connecting securely over an unsecured network

On Linux or macOS (no install required):

`https://www.openssh.com`

On Windows:

`https://putty.org`



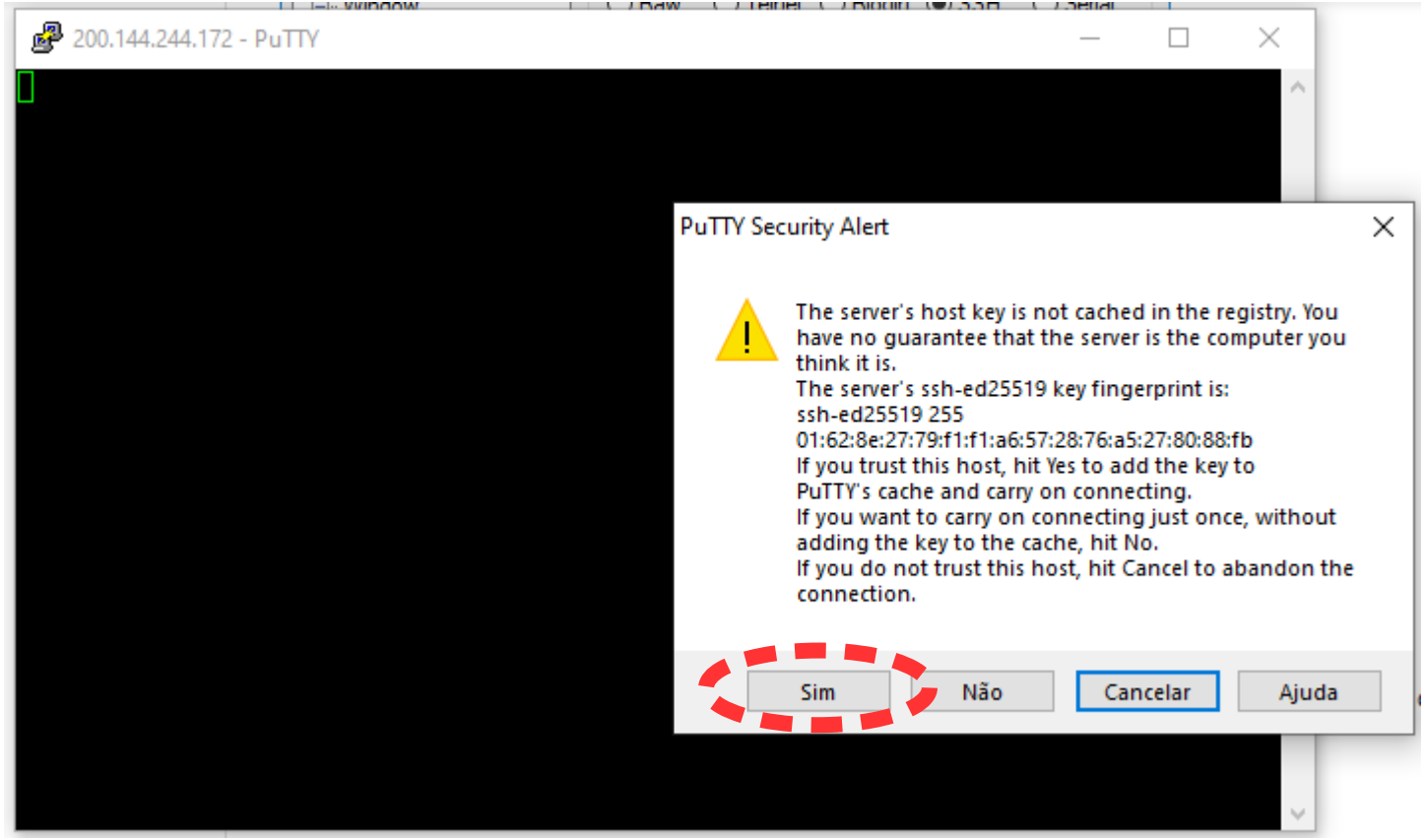
200.144.244.172

Give this connection a name (whatever you want) so you can reuse it later; press **Save**

After you press **Open**, a black screen will appear asking for your user name and, later, password

But wait, there's more...

On your first connection, **and only then**, PuTTY will ask if you trust the remote computer, so it can store the cryptographic key from the remote machine



On your first connection, you will have to change password:

```
jmalves@JDesk:~$ ssh dummy@JLabVM
dummy@200.144.244.172's password:
You are required to change your password immediately (administrator enforced)
Linux JLabVM 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for dummy.
Current password:
New password:
Retype new password:
passwd: password updated successfully
Connection to 200.144.244.172 closed.
jmalves@JDesk:~$
```

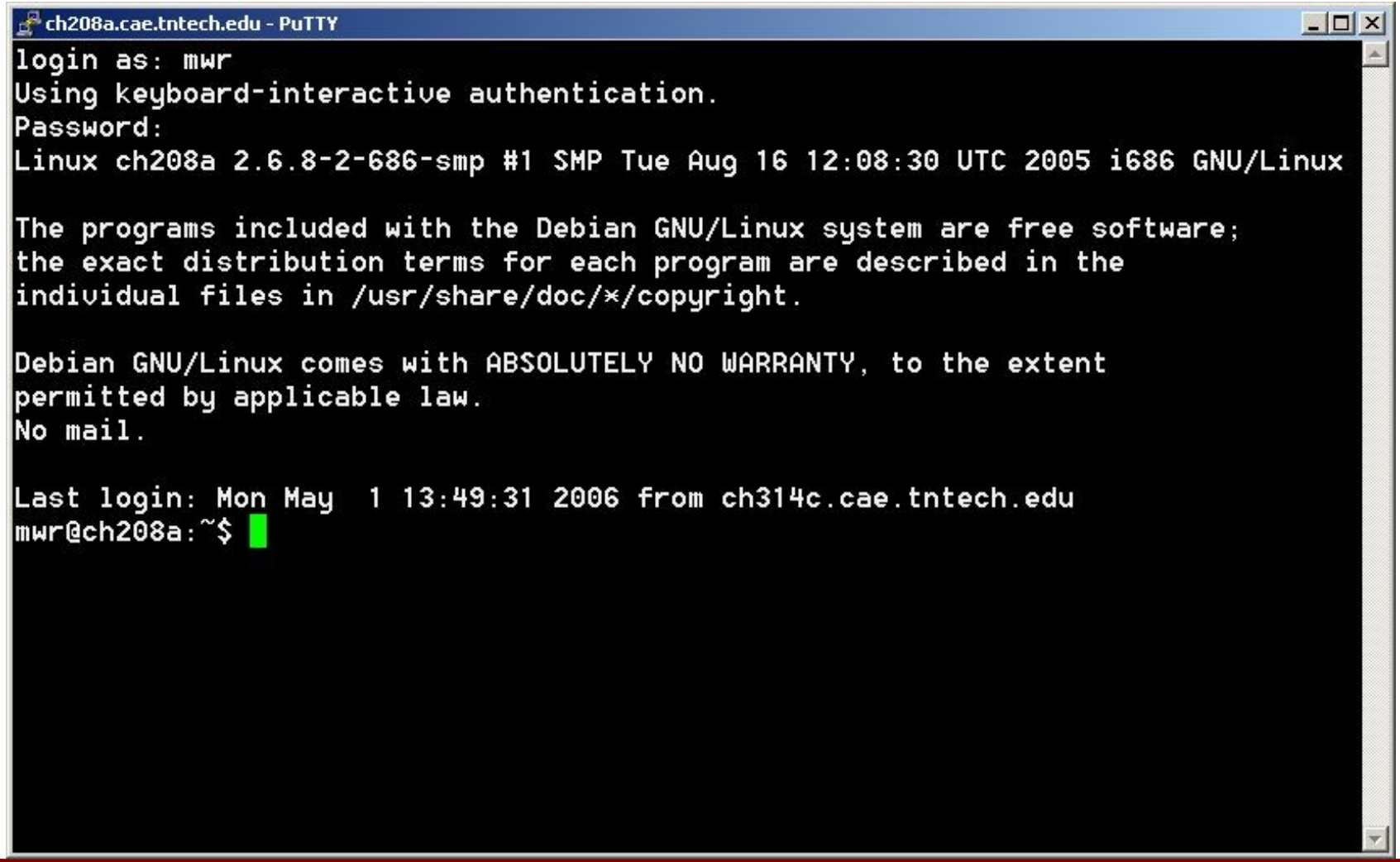
CURRENT first!

X 2

Nothing appears on your screen as you type; this is normal

Your connection will be closed; this is as intended

If your connection is successful, then you see something like this:



But how about Linux?

(or macOS)

Much simpler! No need to install anything (probably).

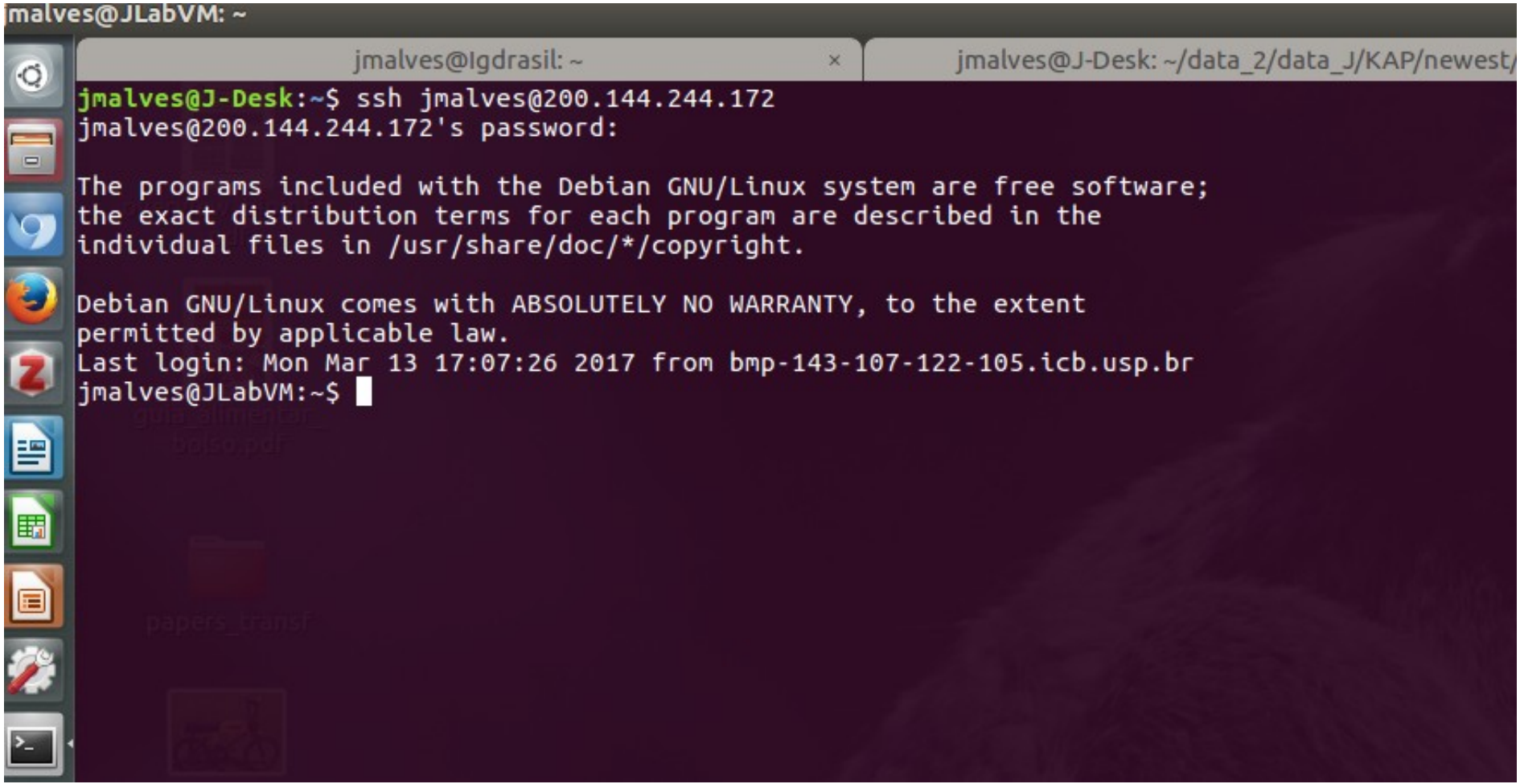
```
ssh user_name@200.144.244.172
```

the **ssh** command

your user name
(in this course, your first name, all lowercase, no accents)

the address of the remote computer (this can be words, e.g., www.google.com, or a numeric IP address)

If your connection is successful, then you see something like this:



On your first connection, you will have to change password:

```
jmalves@JDesk:~$ ssh dummy@JLabVM
dummy@200.144.244.172's password:
You are required to change your password immediately (administrator enforced)
Linux JLabVM 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for dummy.
Current password:
New password:
Retype new password:
passwd: password updated successfully
Connection to 200.144.244.172 closed.
jmalves@JDesk:~$
```

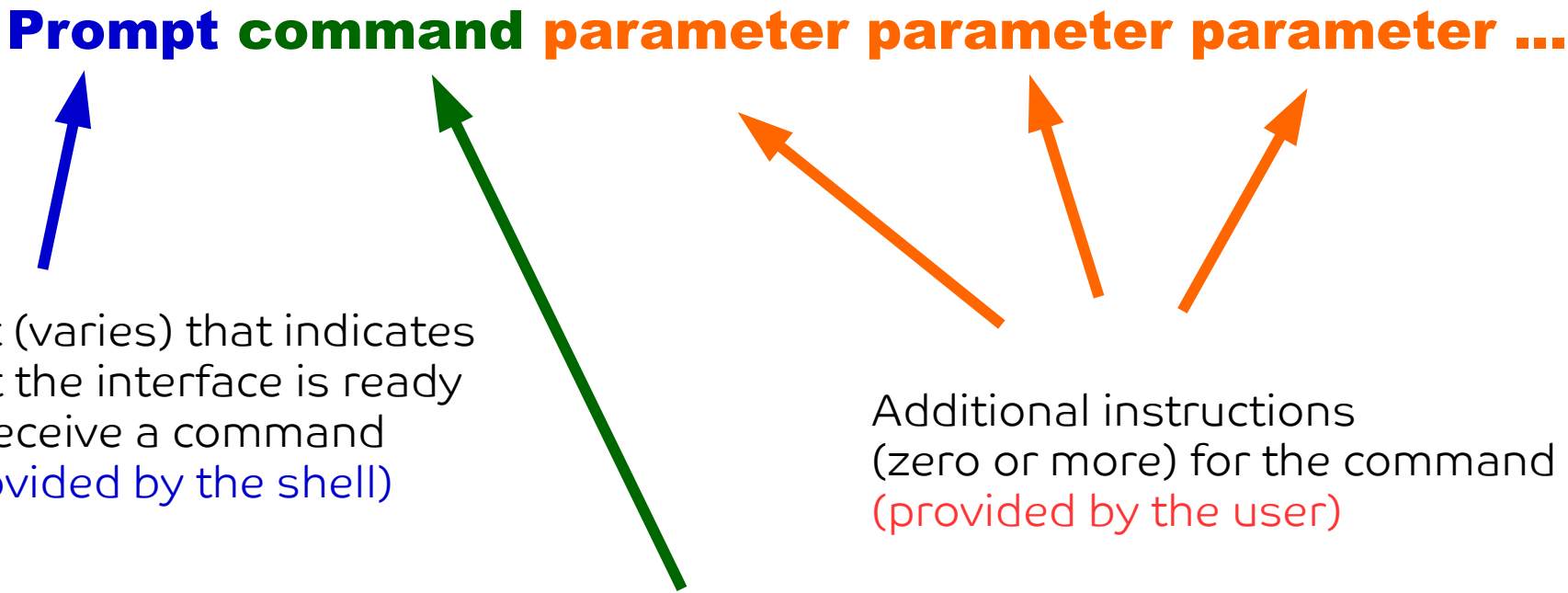
CURRENT first!

X 2

Nothing appears on your screen as you type; this is normal

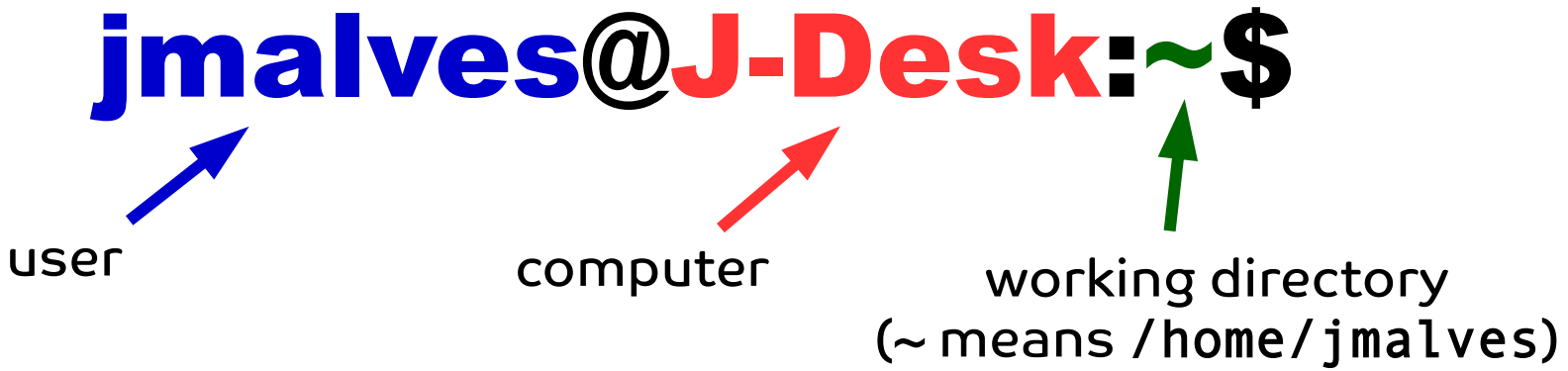
Your connection will be closed; this is as intended

Anatomy of the CLI



The prompt can present a lot of information:

- User name
- Current directory where user is in the file tree
- Computer name
- Whether user is a superuser (root) or not



[jotalves@seal Aamb]$

root@vssh39:~/Docs/assembly#

Quiz time!

Go to the course page and choose **Quiz 1**

Anatomy of the CLI

- A special character marks the end of the prompt: **$** for a regular user, or **#** for the root user
- The command is the name of the program or instruction to be executed
- There are basically two kinds of commands:
  - Built-in: part of the shell itself, it does not depend on an additional program file; for example, **`export`**
  - External: an executable file, not part of the shell, but installed in the operating system; for example, **`ls`**
- Why is that important? How you ask for help depends on whether the command is built-in or external

Anatomy of the CLI

- Depending on the command, arguments might be mandatory (or not)
- Arguments are additional information, sometimes optional, given to the command being run
- Usually, arguments can be of two main kinds:
  - **Switch** or **flag**: it turns an option on or off, like a light switch
  - **Option**: receives a value
- Switches change command behavior in a **binary** way: yes or no
- Options that accept values collect **additional information** like numbers, file or directory names, keywords etc. needed or useful for the command
- The shell is an environment, but also a **simple programming language**

Built-in or external?

How to tell for sure whether something is a built-in command of the shell or an external program?

Use the **`type`** command: if it returns **`X is a shell builtin`**, then it is a built-in command; if a file path is returned, then it's external

A couple of examples:

`type ld`  returns **`ld is /usr/bin/ld`**

`type cd`  returns **`cd is a shell builtin`**

Now you do it!

Go to the Moodle site and enter **Practical Exercise 1**

(after I have activated it)

Follow the instructions to answer the questions in the exercise (which is multiple choice, this time)

Keeping a terminal session alive

- One day, you are happily working on the terminal, accessing a remote machine
- You are running some heavy analysis, which has been going on for hours... (or days!)
- Then, all of a sudden, your **internet connection drops**, or your **laptop crashes**, or you **lose power** at home...
- **You lost all your work!**

Keeping a terminal session alive

- The `screen` program allows us to keep the remote session alive, even when you turn off your local computer

- You can even **reopen the session** from a different computer (say, you start something from work, then go home and access the session from there)

- Most important: everything you were doing is still there (as long as the remote computer does not have problems, of course...)

Keeping a terminal session alive

- To start `screen`, just type its name
- You will see a welcome message, which you can send away pressing the Enter or the space key
- Then... it looks like nothing happened...
- But `screen` is running!
- `screen` also lets you create multiple, uh... screens in your session

Main `screen` commands

- Every `screen` command starts with **`Ctrl+a`**

`Ctr+a c` : create a new screen within the session

`Ctr+a n` : go to next screen in the session (if there is one)

`Ctr+a p` : go to previous screen in the session

`Ctr+a d:` detach the session (goes out of **`screen`**, but keeps the session alive)

`Ctr+a k` : close a screen within the session (same as **`exit`**)

Main `screen` commands

- To reattach a `screen` session later, use **`screen -r`**

- If there is only one session alive, that's all you need; otherwise, you have to give the command the name of the session:

```
screen -r 20904.pts-0.JLabVM
```

```
screen -r 20904
```

- To remotely detach a **`screen`** session:

```
screen -d 20904.pts-0.JLabVM
```

Recap

- Linux, and other Unix or Unix-like systems, are extremely powerful due to their command-line interface
- They are also powerful due to the Unix philosophy (small tools that do one thing well, combined to do something complex)
- Great power = great complexity
- The learning curve for the command-line is admittedly steep, but once you get used to it, you do not settle for anything when large data file processing is what you need

KEEP
CALM
AND
LEARN
LINUX