Linux and Free Software: What and Why?

(Qué son Linux y el Software libre y cómo beneficia su uso a las empresas para lograr productividad económica y ventajas técnicas?)

Jugo Creativo
Universidad de Santander UDES
Bucaramanga, Colombia
7 June 2012

Michael Kerrisk
© 2012
mtk@man7.org
http://man7.org/

Who am I?

- Programmer, educator, and writer
- UNIX since 1987; Linux since late 1990s
- Linux *man-pages* maintainer since 2004
- Author of a book on Linux programming

Overview

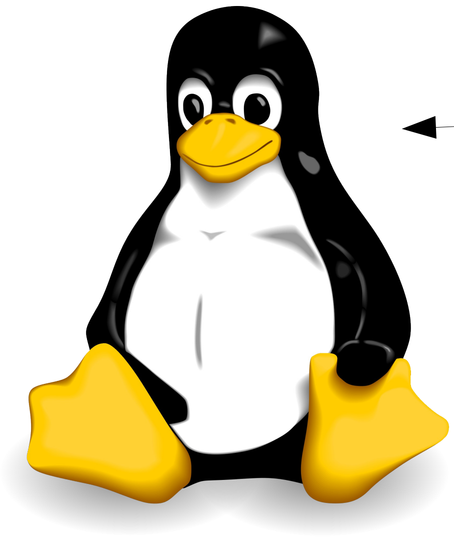
- What is Linux?
- How are Linux and Free Software created?
  - History
- Where is Linux used today?
- What is Free Software?
  - Source code; Software licensing
- **Importance and advantages of Free Software and Software Freedom**
- Concluding remarks

- **What is Linux?**
- How are Linux and Free Software created?
  - History
- Where is Linux used today?
- What is Free Software?
  - Source code; Software licensing
- Importance and advantages of Free Software and Software Freedom
- Concluding remarks

What is Linux?

- An **operating system** (*sistema operativo*)
  - (Operating System = OS)

- Examples of other operating systems:
  - Windows
  - Mac OS X



Penguins are the Linux mascot

But, what's an operating system?

- Two definitions:
  - Kernel
  - Kernel + package of common programs

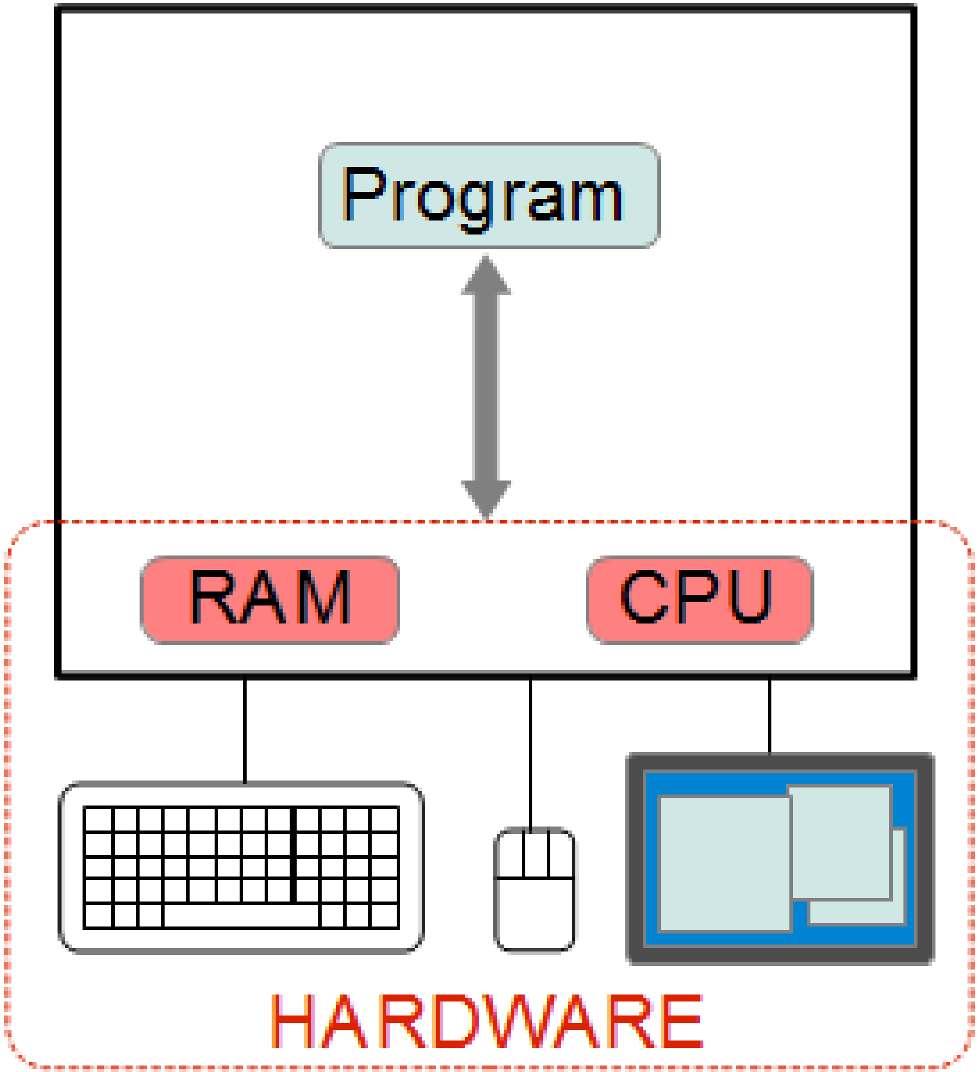
OS Definition 1: Kernel

- Computer scientists' definition:
- Operating System = **Kernel** (*núcleo*)
- Kernel = fundamental program on which all other programs depend

Programs can live without a kernel

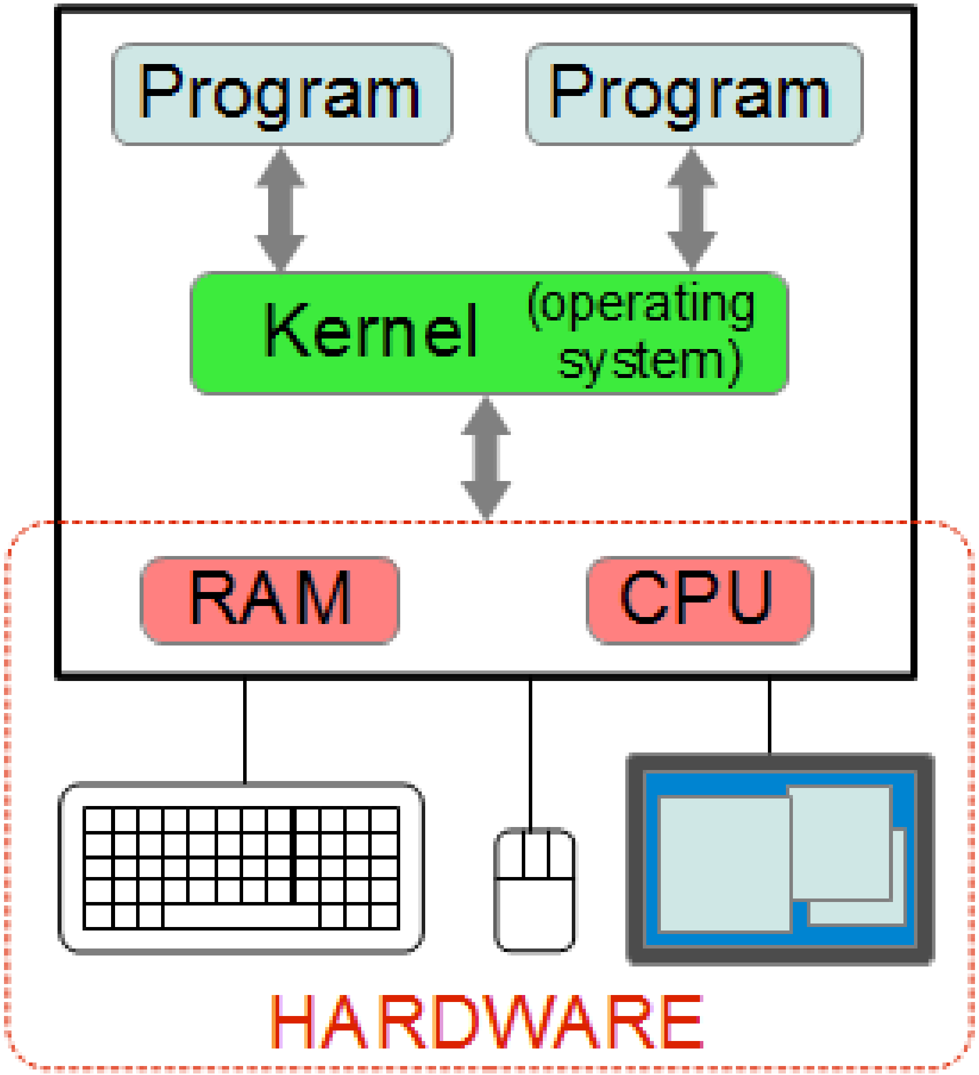
- Programs can run without a kernel

- But, this makes programs limited and complex



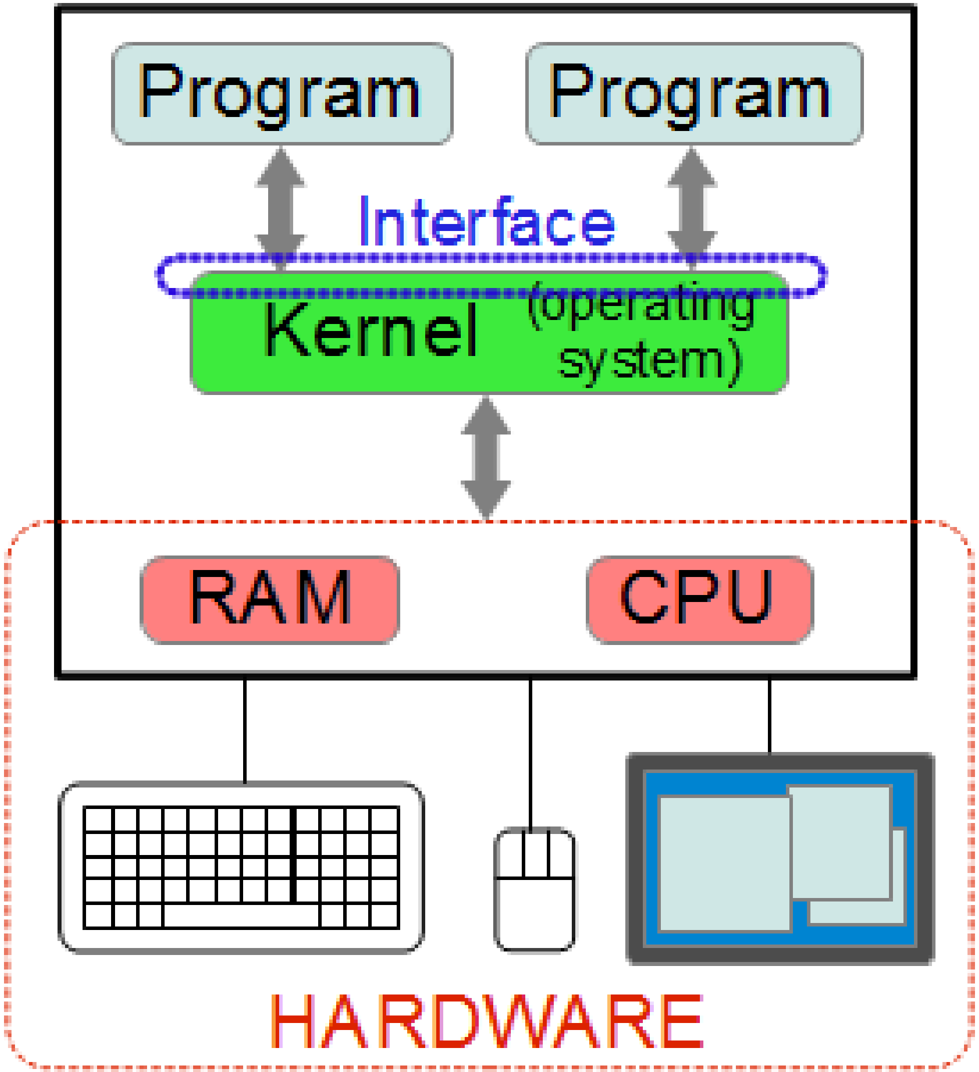
But life is easier with a kernel

- Kernel handles details of different hardware
- Writing programs becomes much easier



Kernel interface

- Programs use standard set of requests to kernel
  - Programming interface (*interfaz de programación*)
- Requests understood by each kernel differ
  - Windows
  - Mac OS X
  - Linux



man7.org

OS Definition 2: Kernel + Packages

- OS = **Kernel + common tools**
- Common/everyday definition of operating system
- Examples of common tools:
  - Graphical user interface (*interfaz gráfica de usuario*)
  - File manager (*explorador de archivos*)
  - Program launcher (*lanzador de programas*)

Linux Operating System

- “**Linux**” is often used to mean either definition
- Some use “**GNU/Linux**” for **kernel+tools**
- “**Linux kernel**” makes distinction clear

- What is Linux?
- **How are Linux and Free Software created?**
    - **History**
- Where is Linux used today?
- What is Free Software?
    - Source code; Software licensing
- Importance and advantages of Free Software and Software Freedom
- Concluding remarks

It helps to start with a little history...

Some history

- 1969
  - UNIX created

  - (Linux is a reimplementation of UNIX)
  - Linus Torvalds was born

Some history

- mid-1980s
  - Richard Stallman (1953-)
  - Initiates GNU project (1984)
    - Create a “free” UNIX system
    - GNU = “GNU's not UNIX”

  - “Principles of software freedom”
    - Creates *Free Software Foundation* (FSF) (1985)

Some history

- 1991
  - Linus Torvalds
  - 21-year old Finnish student
  - Develops first Linux kernel
  - 10,000 lines of source code
  - Runs on Intel x86 (PC)

Time passes

- 1991: Linus asks for help improving the kernel
- 1000s of programmers work on Linux
- Today:
  - 15 million lines of code
  - Linux runs on **many** different types of computers

Back to the main story...

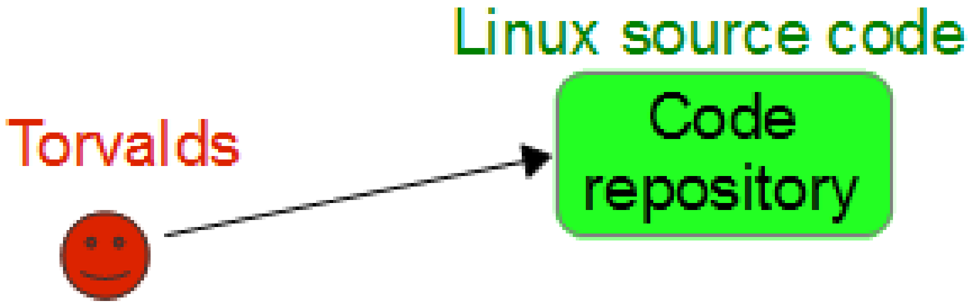
How are Linux and Free Software Created?

Another way of defining Linux

- Linux is:
  - a **very** large software project
  - a collaborative work involving thousands of programmers and hundreds of companies
    - – Collaboration over Internet
      - Email
      - Software tools for exchanging copies of code
- How does it work?

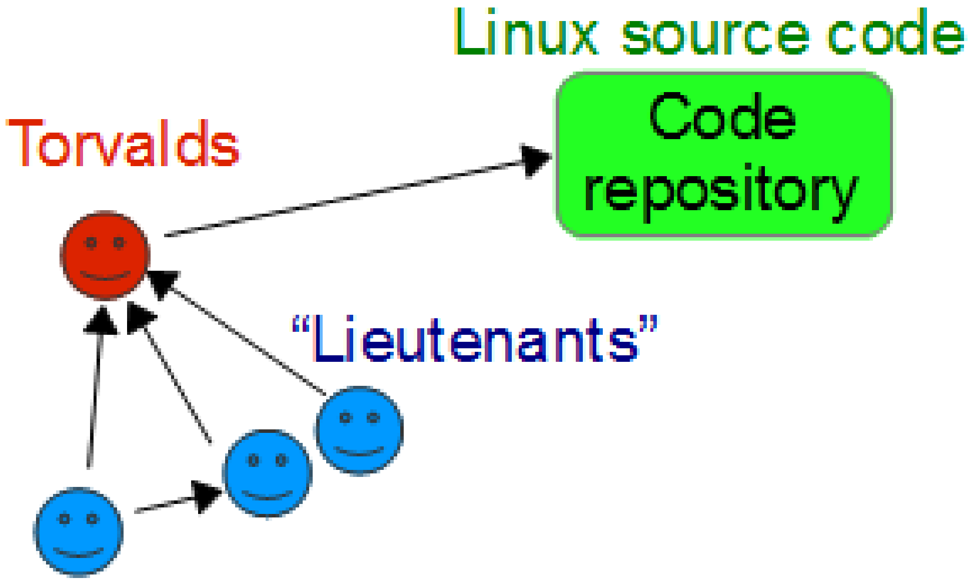
Collaborative development

- Torvalds is gatekeeper (*portero*) of repository (*repositorio*) of Linux code
  - ("Benevolent dictator")



Collaborative development

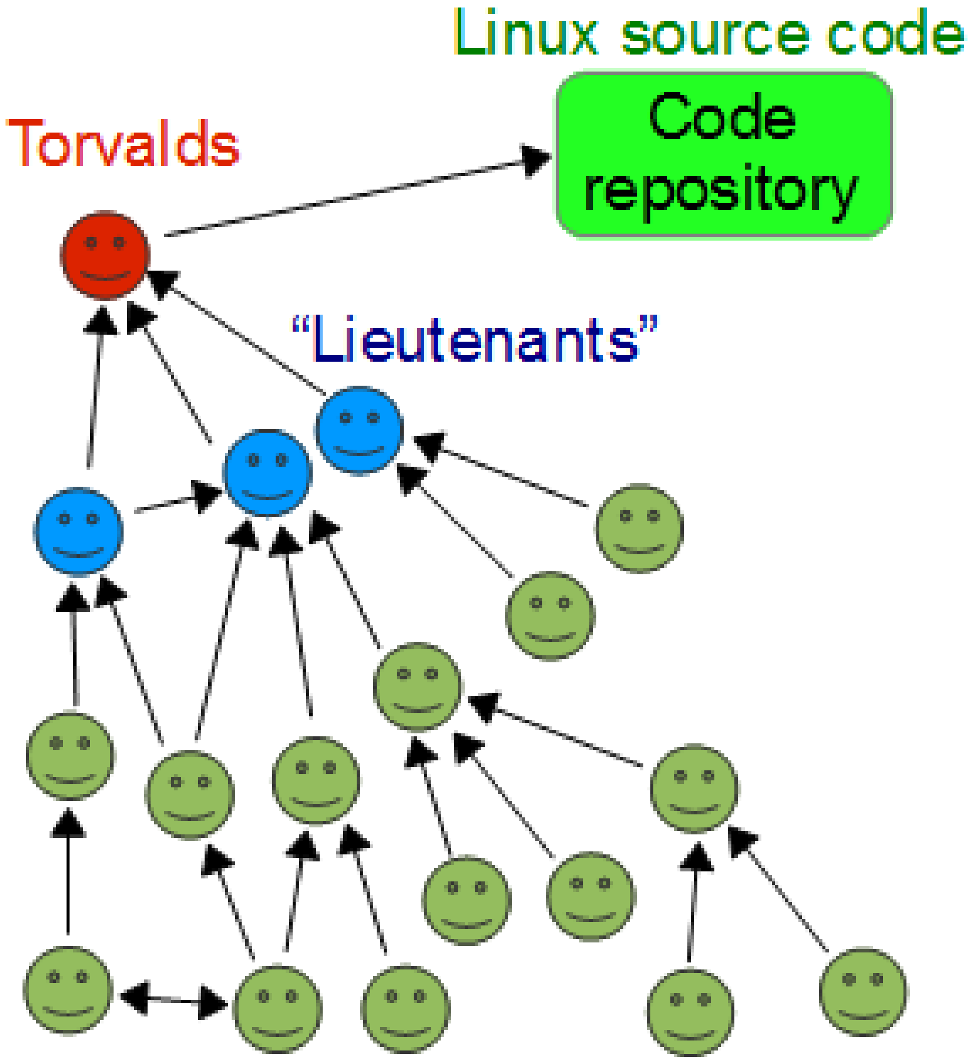
- Torvalds accepts code submitted from "trusted Lieutenants" (*tenientes de confianza*)



- But even having many lieutenants is not enough

Collaborative development

- In turn, lieutenants accept code from others...
- review and improve code and pass to Torvalds



Linux kernel development cycle

- ~ 10-week cycle
- Two-week “merge window”
  - Torvalds takes new code submitted by lieutenants
- Stabilization phase (6-10 weeks)
  - Make all the new stuff actually work
- New stable kernel is *released*
- Next merge window...

A typical kernel release cycle

- Linux 3.4 (see http://lwn.net/Articles/496193/)
  - Released 20 May 2012
  - 9 weeks since Linux 3.3
  - 10,700 separate changes (>7 / hour)
  - 1259 different developers contributed
  - 195 companies contributed
  - “diff” was 1.4 million lines
    - 576k lines added (3.8%)
    - 358k lines removed (2.4%)
      - (Total Linux source code: 15 million lines)

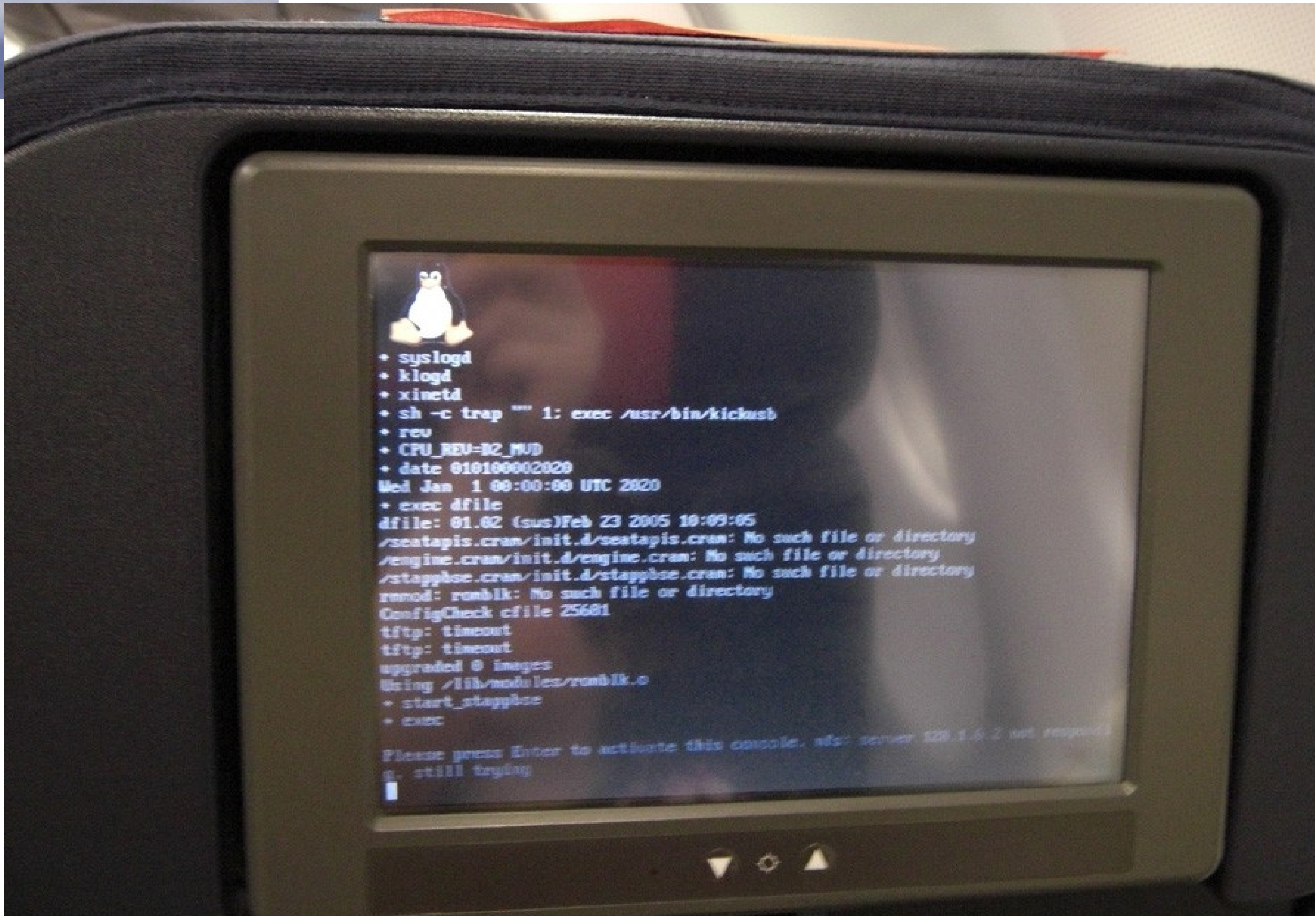
Linux is just one example

- **Thousands** of other Free Software projects with broadly similar development models

LibreOffice
The Document Foundation
VLC
MEDIA PLAYER
gimp
mozilla
Firefox®
APACHE
HTTP SERVER
Audacity®
python™

- What is Linux?
- How are Linux and Free Software created?
    - History
- **Where is Linux used today?**
- What is Free Software?
    - Source code; Software licensing
- Importance and advantages of Free Software and Software Freedom
- Concluding remarks

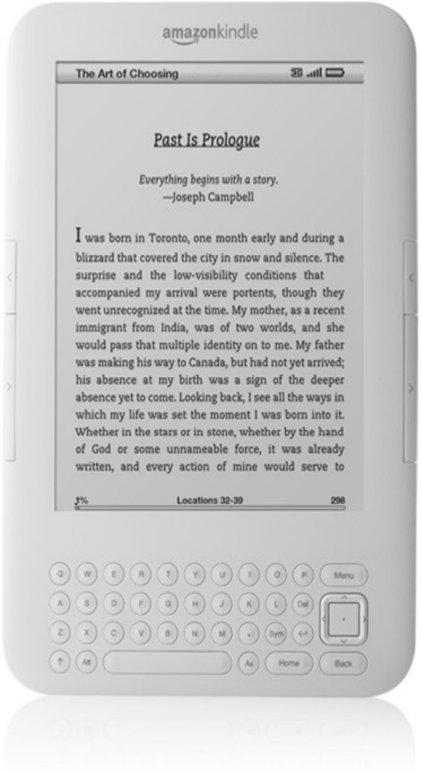
You are already a Linux user...



```
+ syslogd
+ klogd
+ xinetd
+ sh -c trap "" 1; exec /usr/bin/kickusb
+ rev
+ CPU_REV=D2_MVD
+ date 010100002020
Wed Jan  1 00:00:00 UTC 2020
+ exec dfile
dfile: 01.02 (sus)Feb 23 2005 10:09:05
/seatapis.cram/init.d/seatapis.cram: No such file or directory
/engine.cram/init.d/engine.cram: No such file or directory
/stappbse.cram/init.d/stappbse.cram: No such file or directory
rmmod: romblk: No such file or directory
ConfigCheck cfile 25601
tftp: timeout
tftp: timeout
upgraded 0 images
Using /lib/modules/romblk.o
+ start_stappbse
+ exec

Please press Enter to activate this console. nfs: server 120.1.6.2 not respondi
g, still trying
```

Consumer electronics & embedded devices


amazonkindle
The Art of Choosing
Past Is Prologue
Everything begins with a story.
—Joseph Campbell
I was born in Toronto, one month early and during a blizzard that covered the city in snow and silence. The surprise and the low-visibility conditions that accompanied my arrival were portents, though they went unrecognized at the time. My mother, as a recent immigrant from India, was of two worlds, and she would pass that multiple identity on to me. My father was making his way to Canada, but had not yet arrived; his absence at my birth was a sign of the deeper absence yet to come. Looking back, I see all the ways in which my life was set the moment I was born into it. Whether in the stars or in stone, whether by the hand of God or some unnameable force, it was already written, and every action of mine would serve to
1%
Locations 32-39
298
Menu
Home
Back

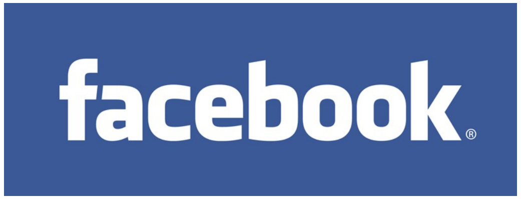
Android

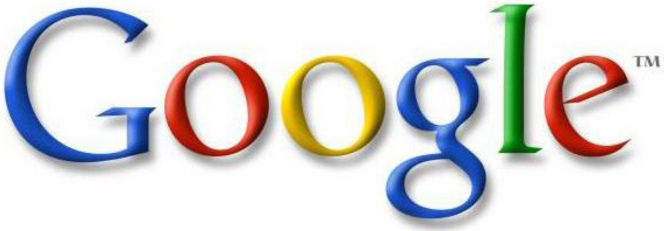
android

SAMSUNG
9:02 AM
3
Phone
Readers Hub
Navigation
Camera
Places
Music
Video
Calendar
Thinkfree
Market
Browser
Applications
Email

4:00
Google
Google+
Google

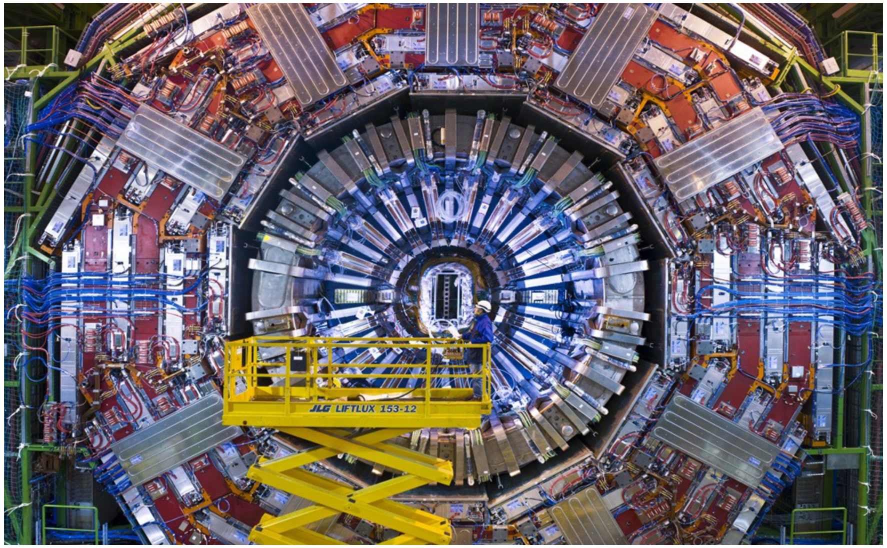
Web services





Web services

And so on...



These enterprises have chosen Linux because they can tailor it to their needs

Where you don't so often find Linux

Linux desktop market share: ~2%

(Mac OS X > 5-10%; MS Windows, 90+%)

Why isn't Linux on the desktop (yet)?

- Once, Linux was not so good at usability
  - Then came Ubuntu...
- People resist change
- Some programs not available on Linux
  - games, Photoshop, QuickBooks, MS Office
- LibreOffice not 100% compatible with MS Office
- PCs usually preinstalled with Windows

- What is Linux?
- How are Linux and Free Software created?
  - History
- Where is Linux used today?
- **What is Free Software?**
  - **Source code; Software licensing**
- Importance and advantages of Free Software and Software Freedom
- Concluding remarks

“Free”

- “Free” creates confusion (in English)
  - “Free as in beer” (*gratis*)
  - “Free as in freedom” (*libertad*)
- Free Software is about **Freedom**
- **Free software ≠ open source software**
  - (More later)

What is software freedom?

- We first need to understand:
  - Source code
  - Software licenses

Code

- Code = set of instructions to a computer:

  → **a program**

- (Nearly) every program exists in two forms...

Source code

- Language used by humans to write programs
  - (*código fuente*)

Machine code

- Language that computers understand
  - (*código de máquina*)

Source code

- High-level language (“English + math”)
  - e.g., C, Java, C++, Python

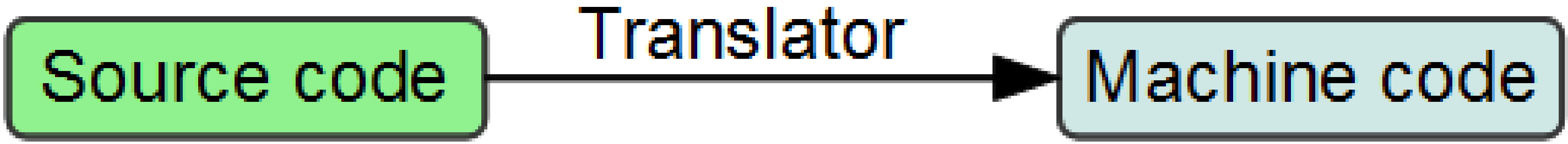
```
int main(void) {
  int j = 0;

  while (j <= 10) {
    j = j + 1;
    printf("%d\n", j);
  }
}
```

Machine code

- Numeric codes
- “Binary code”

```
01110101 11000011
11001001 11001100
11001011 00010001
10101001 01001111
10000101 10100010
10101001 11100111
```

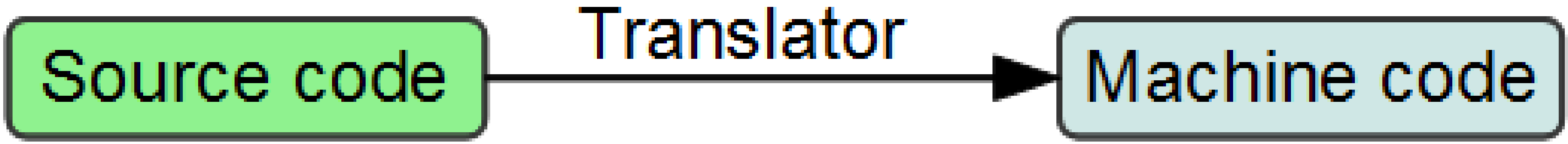


- A translator program converts source code to machine code

```
int main(void) {
  int j = 0;

  while (j <= 10) {
    j = j + 1;
    printf("%d\n", j);
  }
}
```

```
01110101 11000011
11001001 11001100
11001011 00010001
10101001 01001111
10000101 10100010
10101001 11100111
```

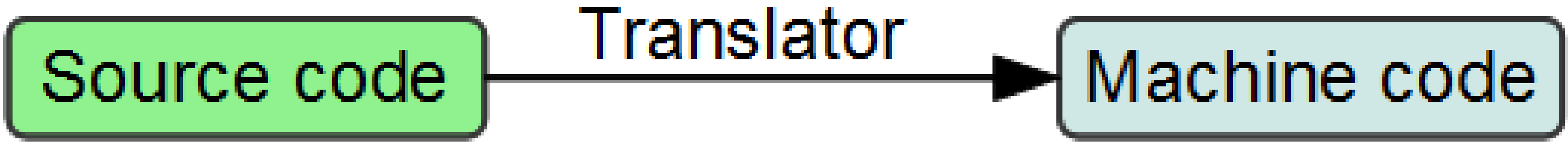


- Easy to learn and write
- Easy to read and change

```
int main(void) {
 int j = 0;
 while (j <= 10) {
  j = j + 1;
  printf("%d\n", j);
 }
}
```

- Hard to learn and write
- Hard to read and change

```
01110101 11000011
11001001 11001100
11001011 00010001
10101001 01001111
10000101 10100010
10101001 11100111
```

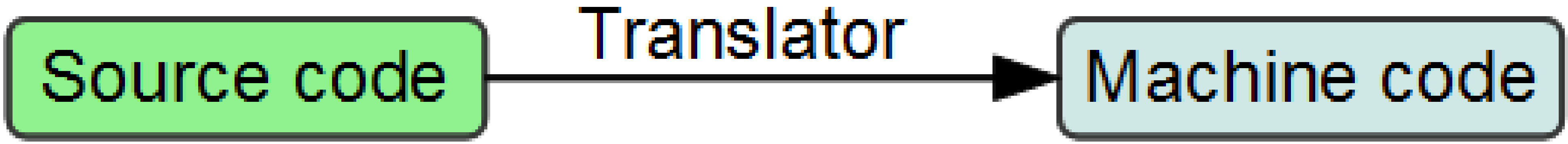


Without source code, it's nearly impossible to change a program

```
int main(void) {
  int j = 0;

  while (j <= 10) {
    j = j + 1;
    printf("%d\n", j);
  }
}
```

```
01110101 11000011
11001001 11001100
11001011 00010001
10101001 01001111
10000101 10100010
10101001 11100111
```



- Free Software gives you source code

- With most commercial software, you get only machine code
  - **“closed source”**

```
int main(void) {
  int j = 0;

  while (j <= 10) {
    j = j + 1;
    printf("%d\n", j);
  }
}
```

```
01110101 11000011
11001001 11001100
11001011 00010001
10101001 01001111
10000101 10100010
10101001 11100111
```

Software licensing

Copyright

- **Copyright** (*derechos de autor*): a set of rights granted to author of a creative work

- Copyright **automatically** applies to any newly created work
  - Author = holder of copyright

Software licensing

- **License** (*licencia*): set of rules used to grant usage rights to others
  - Rights
  - Restrictions
  - Obligations
- License is determined by copyright holder

Proprietary software licenses

- Used by much modern software
  - Windows, MS Office, Photoshop, etc.
- Typical proprietary software producer says:
  - You pay us
  - We'll give you **machine code** version of program
    - **“Closed source”**
  - We give you **limited** rights to use program
- **Free Software licenses are different...**

Free software licenses

Give you freedom

Back to the main story...

What is Free Software?

The software freedoms

- A program is free if (paraphrasing the FSF):

  0) You can **run** the program for any purpose

  1) You can **read** the program and change it as you wish

  2) You can **redistribute** the program to others

  3) You can **improve** the program, **and redistribute** your improved version

- All free software licenses grant these freedoms
  - But they do it differently

Permissive versus reciprocal licenses

Permissive licenses

- Copyright notice
  (*sobre derechos de autor*)
- Grant of rights (the software freedoms)
  (*concesión de derechos*)
- Disclaimer of warranty
  (*descargo de responsabilidad*)

Permissive licenses

- *No obligations*
- You can do just about anything with program...
  - You can improve it, and redistribute only machine-code version
  - “Privatizing” the source code

Reciprocal (copyleft) licenses

- Many kinds, but one is most important
- GNU General Public License (GPL)
- A brilliant creation of Richard Stallman

The GPL

- We give you the source code
- We disclaim warranty
- We grant you the software freedoms
- But, ...

If you distribute
a modified version of the program,
you must distribute
the modified source code

The GPL

- Creates an ever-growing commons (*comunes*) of source code
  - Source code can't be privatized
- Most widely used Free Software license
  - > 50% of all Free Software
  - Linux kernel is licensed under GPL

- What is Linux?
- How are Linux and Free Software created?
  - History
- Where is Linux used today?
- What is Free Software?
  - Source code; Software licensing
- **Importance and advantages of Free Software and Software Freedom**
- Concluding remarks

The software freedoms

- Some obvious advantages from the software freedoms:
  - You can learn by reading and changing the program
  - You can change the program as you wish
  - If the program is broken, you can fix it
  - You can run the program on as many computers as you like
  - You can share the program with others

Free software is cheap

- Free as in freedom ≠ free as in beer, but...

Free software is typically free or cheap

- You may still need to pay for:
  - Training
  - Support
  - Installation and customization
  - Improvements

Free software is auditable

- Is closed source software:
  - Installing extra software you didn't want?
  - Logging your actions?
  - Sending information about you or your actions to “home base”?
- With free software you can **check the source**

Free software is secure

- Free software tends to be more secure
  - Firefox versus Internet Explorer
- Security problems tend to be fixed more quickly
- **Security via audit**
  - You (or someone else) can check the source code
- Opposite is **security via obscurity**...
  - Can you trust vendor's assurance that their software is secure?

“Free” aligns with “openness”

- Free software aligns with **open standards**
  - Open standards allow competition
    - Improved quality
    - Lower costs
- Free software aligns with **open data formats**
  - 20 years from now, will you still be open the file created by your closed source software?

For enterprises

- Can you afford your software infrastructure to be **controlled** by another company?
- Large proprietary software vendor **tell you** what new features you'll get, and when
- If the BSA knocks on your door, can you prove all your proprietary software licenses are in order?

For enterprises

- What happens when your vendor stops supporting the software version you are using?
- Can you change your software supplier?
- What if your supplier goes out of business?

For governments

- See “For enterprises”
- Can you trust closed source software controlled by a (foreign) corporation?
  - Are there “back doors” allowing access to sensitive information?
  - Do you trust a closed source voting machine supplied by a foreign country?
    - Or for that matter, your own government?

- What is Linux?
- How are Linux and Free Software created?
  - History
- Where is Linux used today?
- What is Free Software?
  - Source code; Software licensing
- Importance and advantages of Free Software and Software Freedom
- **Concluding remarks**

“Open source” versus “free”

- Open Source Software
  - “Business friendly” term coined in late 1990s
- Often used as synonyms, but:
  - **Open source** focuses on **practice**: code whose source can be read/modified/redistributed
  - **Free** focuses on **freedom** granted by licenses
- Which term seems more relevant to the advantages?

Some free software you could try now

- All of these run on Linux, Windows, and Max OS X:
  - **LibreOffice**: full-featured Office suite
    - Alternative to MS Office, successor to OpenOffice
  - **Firefox** or **Chrome** web browsers
  - **Thunderbird**: email client
    - **Lightning** adds calendaring
  - **VLC Media Player** (audio + video)
  - **Pidgin**:all your instant messaging in one place
  - **Notepad++**: *Notepad* on steroids
  - **GIMP:** image manipulation (like Photoshop)
  - **Audacity**: audio editing
  - **Blender**: 3D computer graphics
  - **Joomla**: web content creation and management
- And if you haven't already, try a **Linux distribution** (Ubuntu, Fedora, openSUSE, Debian, ...)

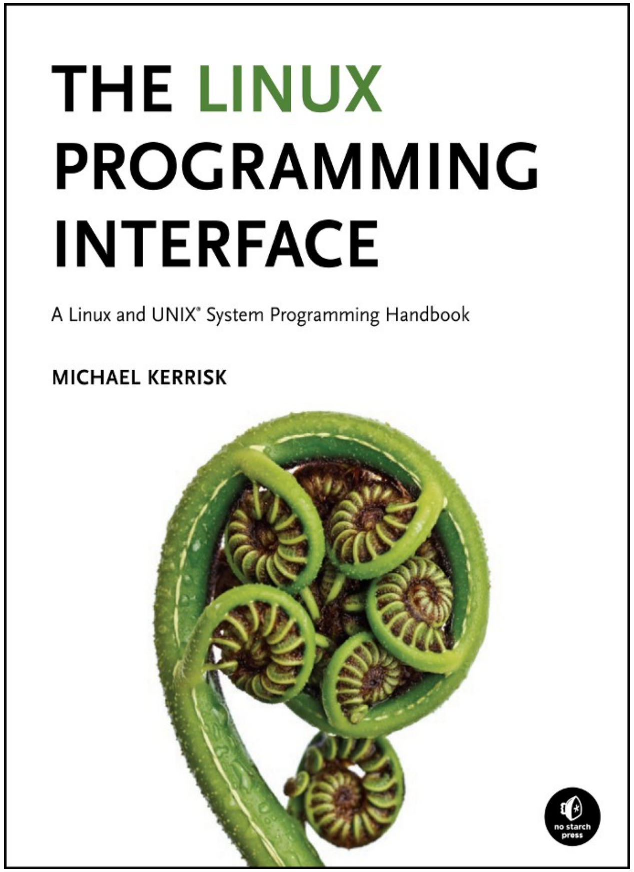
Thanks! And Questions

(Poke me if I forget to repeat anyone's question!)

(slides up soon at http://man7.org/conf/)

Michael Kerrisk
mtk@man7.org
http://man7.org/tlpi

Linux *man-pages* project
mtk.manpages@gmail.com
http://man7.org/linux/man-pages/



THE LINUX PROGRAMMING INTERFACE

A Linux and UNIX® System Programming Handbook

MICHAEL KERRISK

(No Starch Press, 2010)

Mamaku (Black Tree Fern) image (c) Rob Suisted
naturespic.com