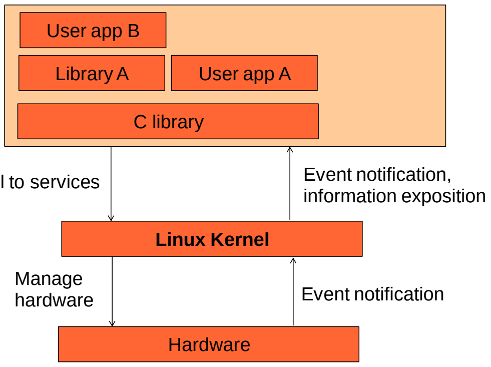
Kernel Linux

Linux kernel in the system



Kernel Linux

- Criado em 1991 como um *hobby* por Linus Torvalds.
- Actualmente é mantido e desenvolvido por centenas de empresas e programadores.
- Desenvolvido em linguagem **C**
- Distribuído em **http://kernel.org**
- Arquitectura modular
  - Módulos podem ser
    - Incluídos estáticamente
    - Carregados dinâmicamente
- Hardware mínimo: 4 Mbytes + processador 32 bit
  - arm, avr32, blackfin, cris, frv, h8300, m32r, m68k, m68knommu, microblaze, mips, mn10300, parisc, s390, sparc, um, xtensa, alpha, ia64, sparc64, powerpc, x86, sh

Linux license

- The whole Linux sources are Free Software released under the GNU General Public License version 2 (GPL v2).
- For the Linux kernel, this basically implies that:
- When you receive or buy a device with Linux on it, you should receive the Linux sources, with the right to study, modify and redistribute them.
- When you produce Linux based devices, you must release the sources to the recipient, with the same rights, with no restriction.
- See http://free-electrons.com/articles/freesw/ training for exact details about Free Software and its licenses.

Linux kernel key features

- **Security**
  It can't hide its flaws. Its code is reviewed by many experts.
- **Stability and reliability**.
- **Modularity**
  Can include only what a system needs even at run time.
- **Easy to program**
  You can learn from existing code. Many useful resources on the net.
- **Portability** and **hardware support**
  Runs on most architectures.
- **Scalability**
  Can run on super computers as well as on tiny devices
  (4 MB of RAM is enough).
- **Compliance to standards** and interoperability.
- Exhaustive **networking** support.

System calls

- The main interface between the kernel and userspace is the set of system calls
- About ~300 system calls that provides the main kernel services
  - File and device operations, networking operations, inter-process communication, process management, memory mapping, timers, threads, synchronization primitives, etc.
- This interface is stable over time: only new system calls can be added by the kernel developers
- This system call interface is wrapped by the C library, and userspace applications usually never make a system call directly but rather use the corresponding C library function

Virtual filesystems

- Linux makes system and kernel information available in user-space through virtual filesystems (virtual files not existing on any real storage). No need to know kernel programming to access such information!
- Mounting /proc:
  mount -t proc none /proc
- Mounting /sys:
  mount -t sysfs none /sys

Filesystem type

Raw device or filesystem image
In the case of virtual filesystems, any string is fine

Mount point

/proc details

A few examples:

- /proc/cpuinfo: processor information
- /proc/meminfo: memory status
- /proc/version: kernel version and build information
- /proc/cmdline: kernel command line
- /proc/<pid>/environ: calling environment
- /proc/<pid>/cmdline: process command line

... and many more! See by yourself!

Lots of details about the /proc interface are available in Documentation/filesystems/proc.txt
(almost 2000 lines) in the kernel sources.

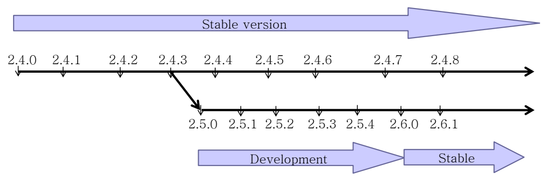
Versões do Kernel Linux

As versões são numeradas com $x.y.z$

- $x.y$: Versão principal (major version)
- z: identifica a versão exacta (minor version)

- Versões estáveis (2.y)
  - y: número par
  - Exemplos: 2.0.40, 2.2.26, 2.4.27, 2.6.7 …
- Versões de desenvolvimento (2.y)
  - y: número impar
  - Exemplos: 2.3.42, 2.5.74

Until 2.6

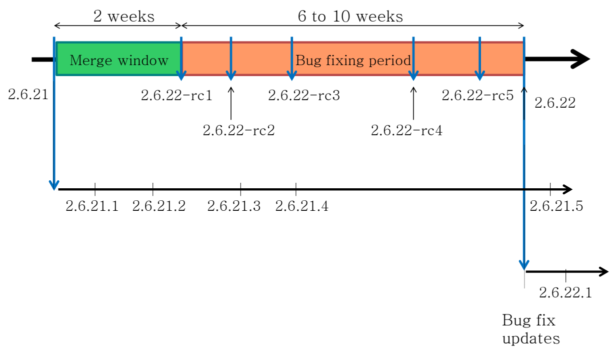


Note: in reality, many more minor versions exist inside the stable and development branches

New 3.x branch

- From 2003 to 2011, the official kernel versions were named 2.6.x.
- Linux 3.0 was released in July 2011
- There is no change to the development model, only a change to the numbering scheme
  - Official kernel versions will be named 3.x (3.0, 3.1, 3.2, etc.)
  - Stabilized versions will be named 3.x.y (3.0.2, 3.4.3, etc.)
  - It effectively only removes a digit compared to the previous numbering scheme

Merge and bug fixing windows



Compilar Linux

- **Obter fontes** de: http://kernel.org/
  - wget http://kernel.org/pub/linux/kernel/v2.6/linux2.6.7.tar.bz2
- **Desempacotar**
  - tar xvjf   /path/to/linux2.6.7.tar.bz2
- **Configurar o Kernel**
  - make xconfig   ou   make menuconfig   ou   make config
- **Compilar o Kernel**
  - make

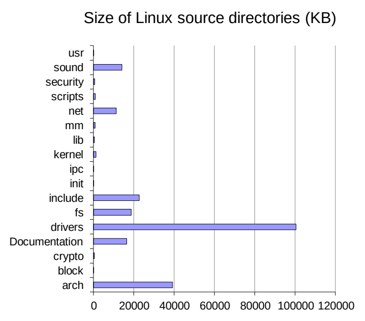
Linux kernel size (1)

- Linux 2.6.31 sources:

  Raw size: 350 MB (30,900 files, approx 12,000,000 lines)
  gzip compressed tar archive: 75 MB
  bzip2 compressed tar archive: 59 MB (better)
  lzma compressed tar archive: 49 MB (best)
- Minimum Linux 2.6.29 compiled kernel size with CONFIG_EMBEDDED, for a kernel that boots a QEMU PC (IDE hard drive, ext2 filesystem, ELF executable support):
  - 532 KB (compressed), 1325 KB (raw)
- Why are these sources so big?

  Because they include thousands of device drivers, many network protocols, support many architectures and filesystems...
- The Linux core (scheduler, memory management...) is pretty small!

Linux kernel size (2)



Linux 2.6.17
Measured with:
du -s --apparent-size

Getting Linux sources

- Full tarballs
  - Contain the complete kernel sources
  - Long to download and uncompress, but must be done at least once
  - Example:
    http://kernel.org/pub/linux/kernel/v2.6/linux-2.6.14.7.tar.bz2
- Incremental patches between versions
  - It assumes you already have a base version and you apply the correct patches in the right order
  - Quick to download and apply
  - Examples
    http://kernel.org/pub/linux/kernel/v2.6/patch-2.6.14.bz2 (2.6.13 to 2.6.14)
    http://kernel.org/pub/linux/kernel/v2.6/patch-2.6.14.7.bz2 (2.6.14 to 2.6.14.7)
- All previous kernel versions are available in
  http://kernel.org/pub/linux/kernel/

diff

original.txt:

```
This part of the document has
stayed the same from version
to version. It shouldn't be
shown if it doesn't change.
Otherwise, that would not be
helping to compress the size
of the changes.

This paragraph contains
text that is outdated.
It will be deleted in the
near future.

It is important to spell check
this dokument. On the other
hand, a misspelled word isn't
the end of the world.
Nothing in the rest of this
paragraph needs to be
changed. Things can be
added after it.
```

diff original.txt new.txt

```
0a1,5
> This is an important notice!
> It should therefore be located
> at the beginning of this
> document!
>
6,12c11
< helping to compress the size
< of the changes.
<
< This paragraph contains
< text that is outdated.
< It will be deleted in the
< near future.
---
> helping to compress anything.
15c14
< this dokument. On the other
---
> this document. On the other
22a22,25
> This paragraph contains
> important new additions
> to this document.
>
```

new.txt:

```
This is an important notice!
It should therefore be located
at the beginning of this
document!

This part of the document has
stayed the same from version
to version. It shouldn't be
shown if it doesn't change.
Otherwise, that would not be
helping to compress anything.

It is important to spell check
this document. On the other
hand, a misspelled word isn't
the end of the world.
Nothing in the rest of this
paragraph needs to be
changed. Things can be
added after it.

This paragraph contains
important new additions
to this document.
```

patch

Criar patch

```
diff   original.txt new.txt   > alteracoes.pat
```

Aplicar Patch

```
patch original.txt   < alteracoes.pat
```

altera original.txt
original.txt = new.txt

Remover Patch

```
patch -R new.txt    < alteracoes.pat
```

altera new.txt
new.txt = original.txt

Anatomy of a patch file

A patch file is the output of the diff command

```
diff -Nru a/Makefile b/Makefile
--- a/Makefile  2005-03-04 09:27:15 -08:00
+++ b/Makefile  2005-03-04 09:27:15 -08:00
@@ -1,7 +1,7 @@
 VERSION = 2
 PATCHLEVEL = 6
 SUBLEVEL = 11
-EXTRAVERSION =
+EXTRAVERSION = .1
 NAME=Woozy Numbat

 # *DOCUMENTATION*
```

← diff command line
← File date info
← Line numbers in files
← Context info: 3 lines before the change
Useful to apply a patch when line numbers changed
← Removed line(s) if any
← Added line(s) if any
← Context info: 3 lines after the change

Using the patch command

The patch command applies changes to files in the current directory:

- Making changes to existing files
- Creating or deleting files and directories

You can reverse a patch with the -R option

patch usage examples:

- patch -p<n> < diff_file
- cat diff_file | patch -p<n>
- bzcat diff_file.bz2 | patch -p<n>
- zcat diff_file.gz | patch -p<n>

You can test a patch with the --dry-run option

n: number of directory levels to skip in the file paths

Applying a Linux patch

Linux patches...

- Always to apply to the x.y.<z-1> version

  Downloadable in gzip
  and bzip2 (much smaller) compressed files.
- Always produced for n=1
(that's what everybody does... do it too!)
- Linux patch command line example:

```
cd linux-2.6.13
bzcat ../patch-2.6.14.bz2 | patch -p1
bzcat ../patch-2.6.14.7.bz2 | patch -p1
cd ..; mv linux-2.6.13 linux-2.6.14.7
```

- Keep patch files compressed: useful to check their signature later.
You can still view (or even edit) the uncompressed data with vim:

  vim patch-2.6.14.bz2 (on the fly (un)compression)

You can make patch 30% faster by using -sp1 instead of -p1 (silent)

Tested on patch-2.6.23.bz2

Kernel configuration (1)

- The kernel contains thousands of device drivers, filesystem drivers, network protocols and other configurable items
- Thousands of options are available, that are used to selectively compile parts of the kernel source code
- The kernel configuration is the process of defining the set of options with which you want your kernel to be compiled
- The set of options depends
  - On your hardware
  - On the capabilities you would like to give to your kernel

Kernel configuration (2)

- The configuration is stored in the .config file at the root of kernel sources
  - Simple text file, key=value style
- As options have dependencies, typically never edited by hand, but through graphical interfaces :
  - make [xconfig|gconfig|menuconfig|oldconfig]
  - These are targets from the main kernel Makefile. Run make help to get a list of all available targets.
- To modify a kernel in a GNU/Linux distribution:
the configuration files are usually released in /boot/, together with kernel images: /boot/config-2.6.17-11-generic

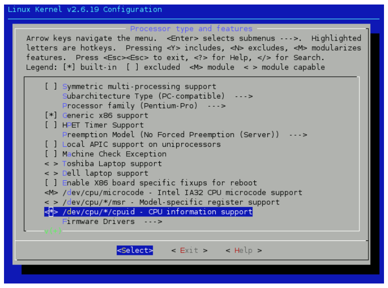
.config file

```
#
# CD-ROM/DVD Filesystems
#
CONFIG_ISO9660_FS=m
CONFIG_JOLIET=y
CONFIG_ZISOFS=y
CONFIG_UDF_FS=y
CONFIG_UDF_NLS=y

#
# DOS/FAT/NT Filesystems
#
# CONFIG_MSDOS_FS is not set
# CONFIG_VFAT_FS is not set
CONFIG_NTFS_FS=m
# CONFIG_NTFS_DEBUG is not set
CONFIG_NTFS_RW=y
…
```

Section name
(helps to locate settings)

All parameters are prefixed with CONFIG_

make menuconfig



```
Linux Kernel v2.6.19 Configuration
                     Processor type and features
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
 letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
 features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.
 Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

     [ ] Symmetric multi-processing support
         Subarchitecture Type (PC-compatible)  --->
         Processor family (Pentium-Pro)  --->
     [*] Generic x86 support
     [ ] HPET Timer Support
         Preemption Model (No Forced Preemption (Server))  --->
     [ ] Local APIC support on uniprocessors
     [ ] Machine Check Exception
     < > Toshiba Laptop support
     < > Dell laptop support
     [ ] Enable X86 board specific fixups for reboot
     <M> /dev/cpu/microcode - Intel IA32 CPU microcode support
     < > /dev/cpu/*/msr - Model-specific register support
     <*> /dev/cpu/*/cpuid - CPU information support
         Firmware Drivers  --->
     v(+)

            <Select>    < Exit >    < Help >
```

make menuconfig

Useful when no graphics are available. Pretty convenient too!

Same interface found in other tools: BusyBox, buildroot...

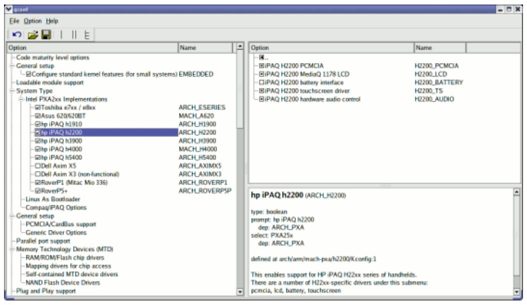
Required Debian packages: libncurses-dev

make xconfig

make xconfig

- The most common graphical interface to configure the kernel.
- Make sure you read
  help -> introduction: useful options!
- File browser: easier to load configuration files
- New search interface to look for parameters
- Required Debian / Ubuntu packages:
  libqt3-mt-dev, g++

make xconfig screenshot



Option	Name
Code maturity level options	
General setup	
☒Configure standard kernel features (for small systems)	EMBEDDED
Loadable module support	
System Type	
Intel PXA2xx Implementations	
☑Toshiba e7xx / e8xx	ARCH_ESERIES
☑Asus 620/620BT	MACH_A620
☑hp iPAQ h1910	ARCH_H1900
☑hp iPAQ h2200	ARCH_H2200
☑hp iPAQ h3900	ARCH_H3900
☑hp iPAQ h4000	MACH_H4000
☑hp iPAQ h5400	ARCH_H5400
☐Dell Axim X5	ARCH_AXIMX5
☐Dell Axim X3 (non-functional)	ARCH_AXIMX3
☑RoverP1 (Mitac Mio 336)	ARCH_ROVERP1
☑RoverP5+	ARCH_ROVERP5P
Linux As Bootloader	
Compaq/iPAQ Options	
General setup	
PCMCIA/CardBus support	
Generic Driver Options	
Parallel port support	
Memory Technology Devices (MTD)	
RAM/ROM/Flash chip drivers	
Mapping drivers for chip access	
Self-contained MTD device drivers	
NAND Flash Device Drivers	
Plug and Play support	

Option	Name
..	
⊞iPAQ H2200 PCMCIA	H2200_PCMCIA
⊞iPAQ H2200 MediaQ 1178 LCD	H2200_LCD
☐iPAQ H2200 battery interface	H2200_BATTERY
⊞iPAQ H2200 touchscreen driver	H2200_TS
⊞iPAQ H2200 hardware audio control	H2200_AUDIO

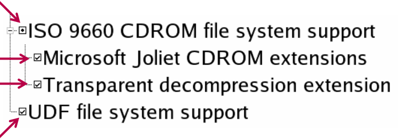
hp iPAQ h2200 (ARCH_H2200)

```
type: boolean
prompt: hp iPAQ h2200
  dep: ARCH_PXA
select: PXA25x
  dep: ARCH_PXA

defined at arch/arm/mach-pxa/h2200/Kconfig:1

This enables support for HP iPAQ H22xx series of handhelds.
There are a number of H22xx-specific drivers under this submenu:
pcmcia, lcd, battery, touchscreen
```

Kernel configuration options

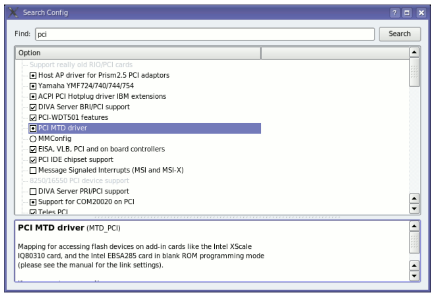
Compiled as a module (separate file)
CONFIG_ISO9660_FS=m

Driver options
CONFIG_JOLIET=y
CONFIG_ZISOFS=y


ISO 9660 CDROM file system support
Microsoft Joliet CDROM extensions
Transparent decompression extension
UDF file system support

Compiled statically into the kernel
CONFIG_UDF_FS=y

make xconfig search interface



Option
Support really old RIO/PCI cards
⊡ Host AP driver for Prism2.5 PCI adaptors
⊡ Yamaha YMF724/740/744/754
⊡ ACPI PCI Hotplug driver IBM extensions
☑ DIVA Server BRI/PCI support
☑ PCI-WDT501 features
⊡ PCI MTD driver
○ MMConfig
☑ EISA, VLB, PCI and on board controllers
☑ PCI IDE chipset support
☐ Message Signaled Interrupts (MSI and MSI-X)
8250/16550 PCI device support
☐ DIVA Server PRI/PCI support
⊡ Support for COM20020 on PCI
☑ Teles PCI

PCI MTD driver (MTD_PCI)

Mapping for accessing flash devices on add-in cards like the Intel XScale IQ80310 card, and the Intel EBSA285 card in blank ROM programming mode (please see the manual for the link settings).

Looks for a keyword in the description string

Allows to select or unselect found parameters.

Kernel option dependencies

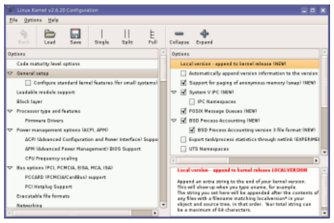
- There are dependencies between kernel options
- For example, enabling a network driver requires the network stack to be enabled
- Two types of dependencies
  - *depends on* dependencies. In this case, option A that depends on option B is not visible until option B is enabled
  - *select* dependencies. In this case, with option A depending on option B, when option A is enabled, option B is automatically enabled
  - *make xconfig* allows to see all options, even those that cannot be selected because of missing dependencies. In this case, they are displayed in gray

make gconfig

make gconfig

New GTK based graphical configuration interface. Functionality similar to that of make xconfig.

Just lacking a search functionality.



```
Options
    Code maturity level options
▽   General setup
        ☐ Configure standard kernel features (for small systems)
    Loadable module support
    Block layer
▽   Processor type and features
        Firmware Drivers
▽   Power management options (ACPI, APM)
        ACPI (Advanced Configuration and Power Interface) Suppo
        APM (Advanced Power Management) BIOS Support
        CPU Frequency scaling
▽   Bus options (PCI, PCMCIA, EISA, MCA, ISA)
        PCCARD (PCMCIA/CardBus) support
        PCI Hotplug Support
    Executable file formats
    Networking
```

```
Options
    Local version - append to kernel release (NEW)
    ☐ Automatically append version information to the version
    ☑ Support for paging of anonymous memory (swap) (NEW)
▽   ☑ System V IPC (NEW)
        ☐ IPC Namespaces
    ☑ POSIX Message Queues (NEW)
▽   ☑ BSD Process Accounting (NEW)
        ☑ BSD Process Accounting version 3 file format (NEW)
    ☐ Export task/process statistics through netlink (EXPERIME
    ☐ UTS Namespaces
```

Local version - append to kernel release LOCALVERSION

Append an extra string to the end of your kernel version.
This will show up when you type uname, for example.
The string you set here will be appended after the contents of
any files with a filename matching localversion* in your
object and source tree, in that order. Your total string can
be a maximum of 64 characters.

Required Debian packages: libglade2-dev

make oldconfig

make oldconfig

- Needed very often!
- Useful to upgrade a .config file from an earlier kernel release
- Issues warnings for configuration parameters that no longer exist in the new kernel.
- Asks for values for new parameters

If you edit a .config file by hand, it's strongly recommended to run make oldconfig afterwards!

make allnoconfig

make allnoconfig

- Only sets strongly recommended settings to y.
- Sets all other settings to n.
- Very useful in embedded systems to select only the minimum required set of features and drivers.
- Much more convenient than unselecting hundreds of features one by one!