Compilación del Kernel

MCC Gerardo Contreras Vega
puntog@gmail.com

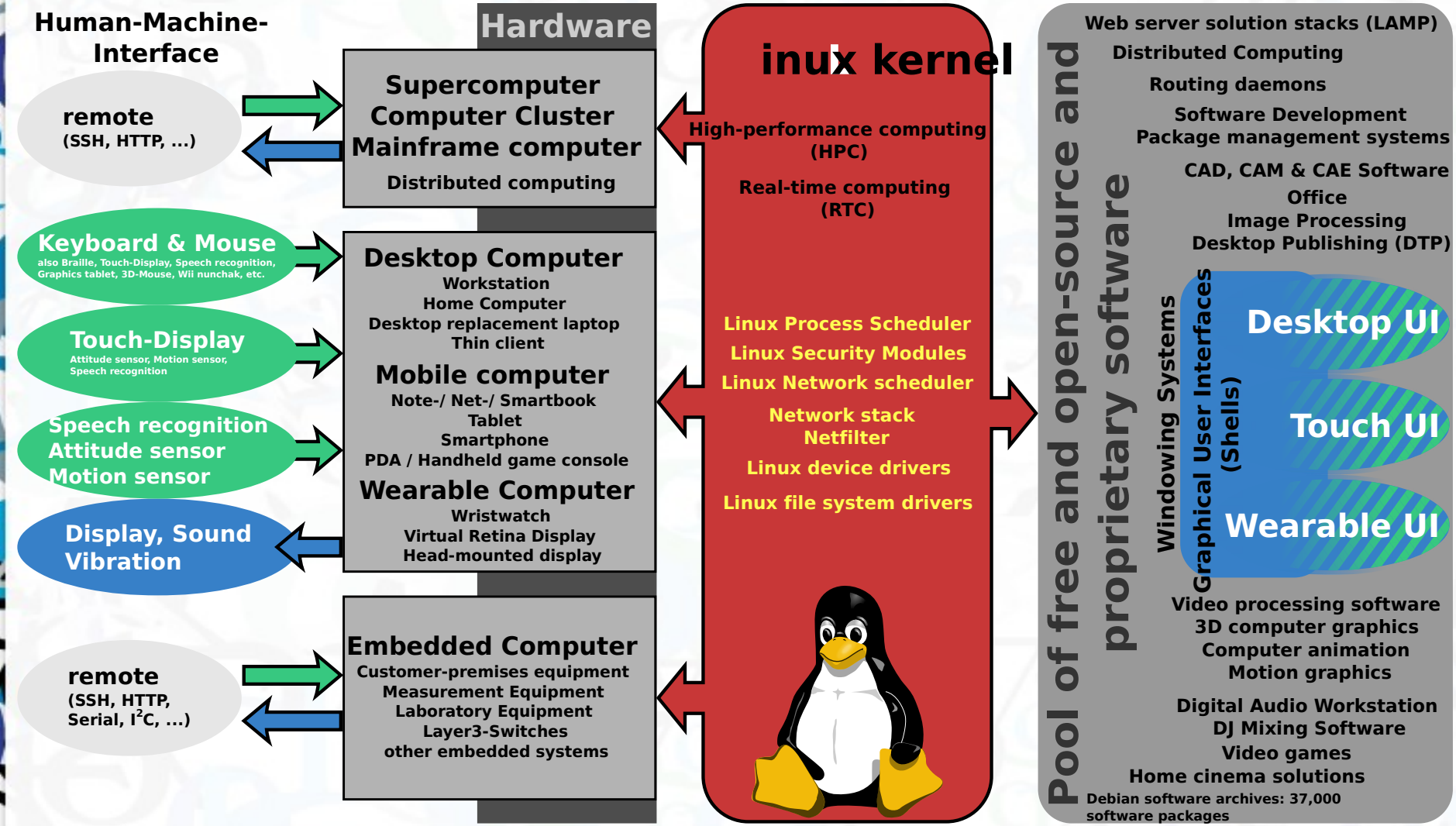
Kernel de GNU/Linux

- El Kernel Linux es el núcleo tipo Unix del Sistema Operativo GNU/Linux.
- Fue desarrollado por Linus B. Torvals en marzo 1991.

```
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 [...] Yes - it's free of any minix code, and it has a multi-
threaded fs. It is NOT portable (uses 386 task switching etc), and it
probably never will support anything other than AT-harddisks, as that's
all I have :-(. [...] It's mostly in C, but most people wouldn't call what
I write C. It uses every conceivable feature of the 386 I could find, as
it was also a project to teach me about the 386. As already mentioned, it
uses a MMU, for both paging (not to disk yet) and segmentation. It's the
segmentation that makes it REALLY 386 dependent (every task has a 64Mb
segment for code & data - max 64 tasks in 4Gb. Anybody who needs more than
64Mb/task - tough cookies). [...] Some of my "C"-files (specifically mm.c)
are almost as much assembler as C. [...] Unlike minix, I also happen to
LIKE interrupts, so interrupts are handled without trying to hide the
reason behind them
```

Kernel Linux

- En septiembre 1991 Linux ver 0.01 10,239 líneas
- Febrero 1992 Linux adopta la licencia GPL
- 14 de Marzo 1994, version 1.0 150,867 líneas
- 3 de noviembre 2009, versión 2.6.32, 9'771,588 líneas
- Versión 4.5.4 16'980,546 líneas
- ***Funny Statistics for the Linux Kernel; https://www.linuxcounter.net/statistics/kernel, visitada noviembre 2016***



Características

- Kernel Monolítico
- Escrito en lenguaje C
- Módulos cargables en el núcleo
- Portable
- Version Vanilla
- Version A.B.C[.D]
  - A Versión
  - B Revisión mayor
  - C Revisión menor
  - D Error grave

¿Por qué compilar?

- Suporte hardware mas reciente
- Personalizar
- Optimizar
- Seguridad
- Aprendizaje

Actividad

- Investigar:
  - Principales desarrolladores del kernel
  - Versión estable más reciente
  - Buscar y analizar el “Linux Kernel Map”

Compilación del kernel, Debian way

- Instalar:
  - kernel-package
  - git
  - libncurses-dev
  - fakeroot
  - wget
  - bzip2
  - build-essential
  - xz-utils
- Tiempo
- Al menos 10Gb espacio

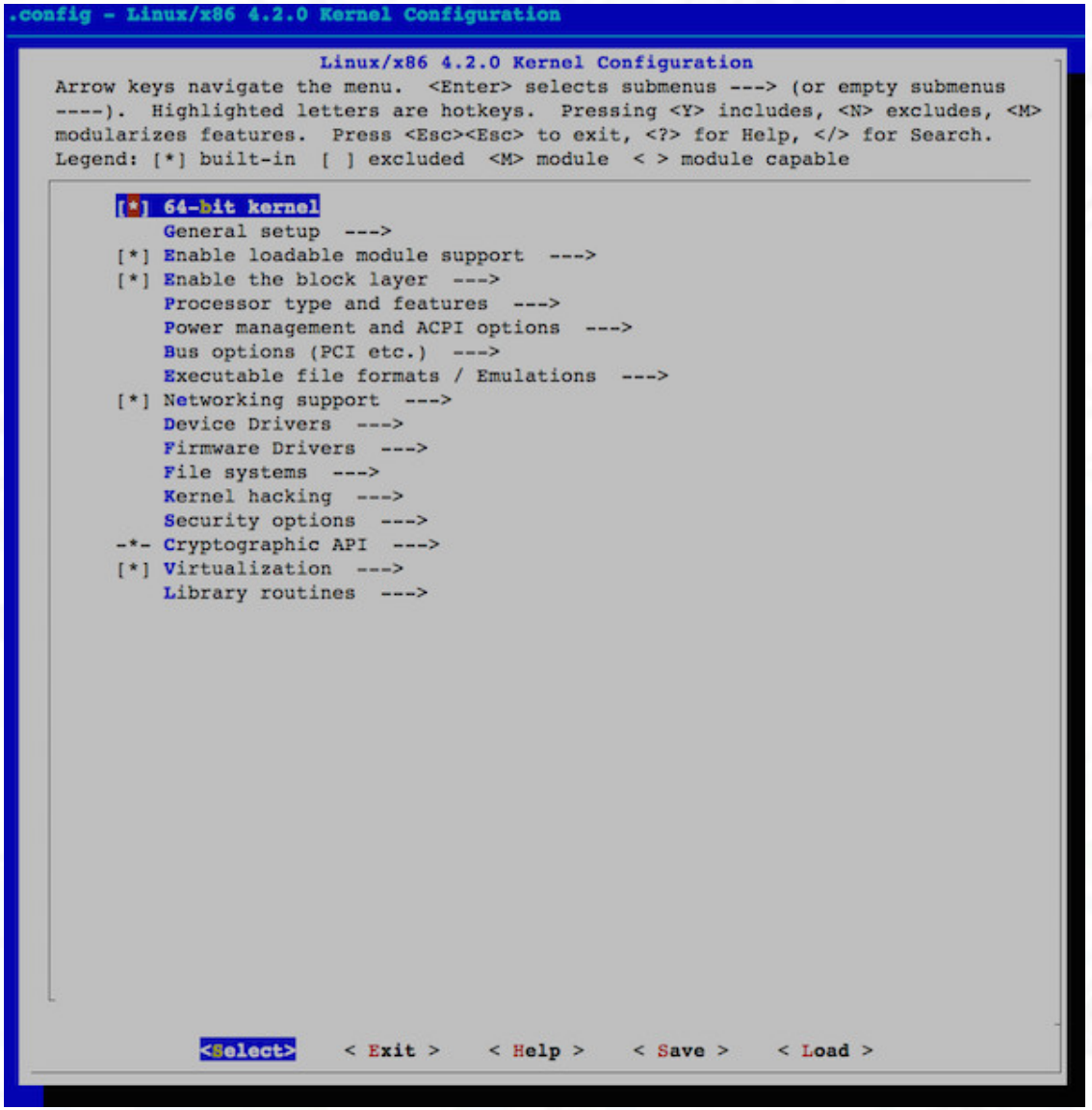
Compilación al estilo Debian(2)

- Descargar kernel http://www.kernel.org
- Descomprimirlo en /usr/src
  - tar xvJf kernel-version.tar.xz
- Crear enlace simbólico (opcional)
  - ln -s kernel-version linux

Compilación del Kernel

- Aplicar parches (opcional)
  - `bzip2 -dc /usr/src/parche.bz2 | patch -p1`
  - `patch -p1 < archivo.patch`
- Configurar el kernel
  - `make-kpkg clean`
  - Si se quiere cargar las opciones del kernel actual
    - `cp /boot/config-`uname -r` ./.config`
  - `make menuconfig`

Explicación de elegir opciones del Kernel



Después de configurar

- `fakeroot make-kpkg –initrd \`

  `--append-to-version “-minucleo” \`

  `kernel_image kernel_headers`
- En /usr/src se crean dos archivos .deb
- Instalarlos
  - `dpkg -i linux-image-version.deb`
  - `dpkg -i linux-headers-version.deb`
- grub se actualiza automáticamente
- Reiniciar el sistema

Compilación del Kernel a la manera tradicional

- Instalar
  - kernel-package
  - libncurses5-dev
  - fakeroot
  - wget
  - bzip2
  - build-essential

Compilación del Kernel

- Descargar Kernel de www.kernel.org
- Descomprimir código en /usr/src
- Aplicar parches al Kernel (opcional)
  - `bzip2 -dc /usr/src/parche.bz2 | patch -p1`
  - `bzip2 -dc /usr/src/parche.bz2 | patch -p1`

Compilación del Kernel a la manera tradicional

- Configurar el Kernel
  - `make clean`
  - `make mrproper`
  - `cp /boot/config-`uname -r` ./.config`
  - `make menu config`
- Construir el kernel
  - `make all`
  - `make modules_install`
  - `make install`

Compilación

- Después de instalar
  - `depmod version-kernel`
  - `apt-get install yaird`

Después de configurar

- `mkinitrd.yaird -o /boot/initrd.img-version version`
- Actualizar grub
  - `update-grub`
- Reiniciar el sistema

Ejercicio

- Obtener información del hardware de la máquina con hwinfo (si no se encuentra, instalarlo)
- Compilar un Kernel al estilo Debian