Introducción a GNU-Linux

para Informáticos

Jon Valdés

juanval @ gmail . com

Unai Aguilera

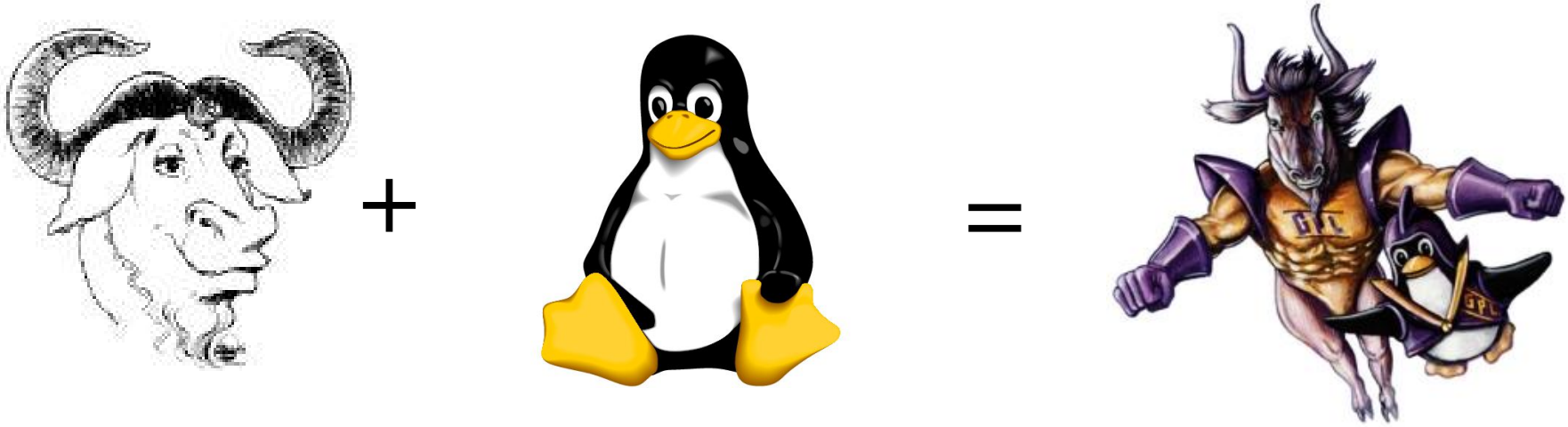
gkalgan @ gmail . com

Cursillos de Julio 2006

Indice

¿Qué es GNU/Linux?

- GNU/Linux es un S.O formado por la unión del proyecto GNU y el núcleo Linux.
  - Esta basado en UNIX (UNIX -Like)



Un poco de historia

- Unix es un S.O desarrollado en los años 70 por Ken Thompsom y Dennis Ritchie (AT&T)
- Se diseñó para ser
  - Portable
  - Multitarea
  - Multiusuario (tiempo compartido)
- Se convirtió en el principal S.O y tuvo un gran impacto en el desarrollo de la computación y de lo que es Internet.

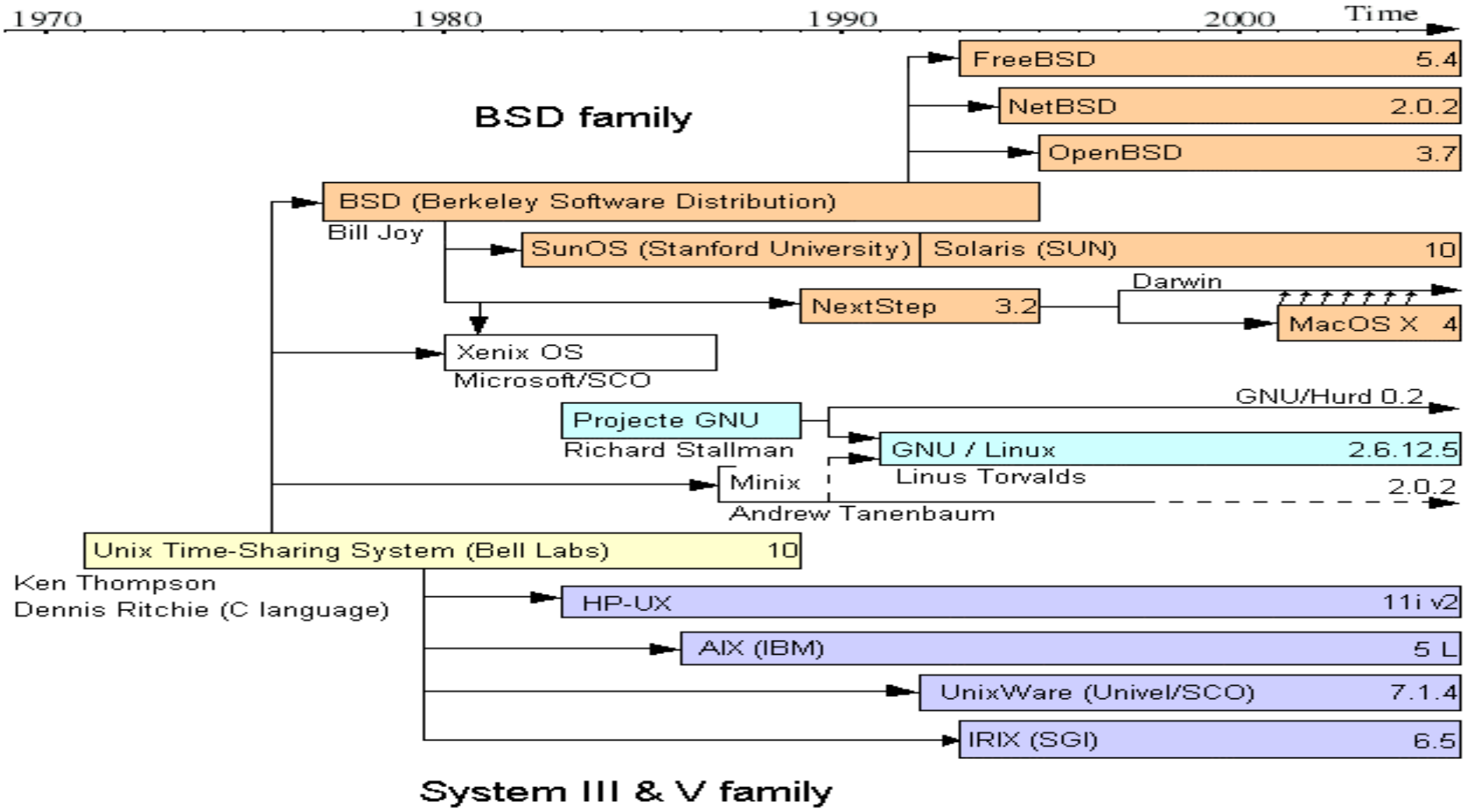
Un poco de historia

- Características principales de los S.O Unix
  - Ficheros de configuración de texto en claro
  - Interprete de comandos
  - Sistema de ficheros jerárquico
  - Gestión de usuarios y acceso.
  - Tratar los dispositivos y la comunicación entre procesos como ficheros
  - Esta constituido por programas independientes, sencillos y por lo tanto fácilmente mantenibles e intercambiables
  - Lenguaje C (desarrollado para crear UNIX)

Un poco de historia

- UNIX era distribuido mediante unas licencias de uso (universidades, gobierno, grandes empresas)
- Se crean variaciones del UNIX original (hasta los años 80 AT&T distribuía el código fuente)
  - BSD creado en la Universidad de Berkeley
  - HP-UX creado por Hewllet-Packard
  - AiX de IBM
  - SunOS (luego llamado Solaris) de Sun Microsystems
  - ...

Un poco de historia



Una poco de historia

- Durante esta estos primeros años los programadores formaban grupos pequeños en las universidades
  - Si alguien hacía una mejora la compartía con el resto de la comunidad y todo el mundo salía beneficiado
  - Era el denominado espíritu *hacker* de las universidades de los años 60 y 70.
- Sin embargo, esto comenzó a cambiar a principio de los años 80

Un poco de historia

- Se empezaron a crear muchas compañías de software que contrataron a los investigadores de las universidades
  - Estas compañías empezaron a comercializar el software utilizando una serie de licencias restrictivas
    - A pesar de que algunos eran programas que habían sido desarrollados inicialmente desde universidades y de forma conjunta colaborativa

Un poco de historia

- En los contratos de venta del software (licencias) incluyeron ciertas restricciones
  - Algunas para impedir que el usuario pueda modificar o saber como funciona un programa que esta usando
    - Además el código fuente ya no se distribuye con el programa por lo que realmente es imposible modificar nada.
  - Richard Stallman que fue uno de esos hackers (trabajaba en el MIT desde los 70) no estaba de acuerdo con esta nueva situación y buscó una solución.

GNU

- Stallman no quería utilizar en su trabajo software que le limitara
  - Pensó que lo que debía hacer es crear un S.O compatible con UNIX pero sin las limitaciones de licencia. Un S.O **libre** de restricciones.
    - Primeramente dejo su trabajo en el MIT para evitar que pudieran reclamar parte de sus futuros programas
    - Ideó una licencia que permitiese esta libertad y que garantizase que el software es y sería siempre libre. Evitando así apropiaciones indebidas por parte de terceros.

La licencia GPL

- Licencia GPL (GNU Public License)
  - Libertad 0: La libertad de usar el programa, con cualquier propósito.
  - Libertad 1: La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades.
  - Libertad 2: La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
  - Libertad 3: La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Proyecto GNU

- GNU significa
  - *GNU is not UNIX*
- El proyecto de crear un S.O por su cuenta era un proyecto complejo
  - Recibió la ayuda de programadores de todo el mundo y en conjunto desarrollaron casi todo el S.O
  - Partieron de un UNIX comercial y fueron substituyendo partes hasta que solo quedó el núcleo

Aparece Linux

- Linux es un núcleo de UNIX desarrollado por Linux Torvalds en el año 1991 y mejorado por gente de todo el mundo desde entonces
  - Torvalds distribuyó su núcleo con la licencia GPL y por eso fue usado para completar el sistema operativo GNU
  - Linux es “solo” el núcleo del sistema operativo y uno de los posibles:
    - GNU + Linux = GNU/Linux
    - El núcleo oficial del proyecto GNU se llama Hurd (GNU/Hurd)

GNU/Linux, soporte

- GNU/Linux funciona actualmente en más de 17 arquitecturas diferentes:
  - En PCs desde 386 hasta los últimos Pentium o AMD de 64 bits.
  - En PowerPC, los procesadores de los Macs.
  - En sistemas menos domésticos, como procesadores Alpha, SPARC, etc.
  - En sistemas embebidos, como lavadoras, microondas, etc.

GNU/Linux, distribuciones

- GNU/Linux normalmente se distribuye en colecciones de software que se llaman precisamente “distribuciones”.
- Una distribución suele tener el Sistema Operativo y más programas, como navegadores, programas de ofimática, juegos, etc.
- Ejemplos de distribuciones conocidas: Red Hat, Mandrake, SuSE, Debian, Gentoo, Ubuntu…

GNU/Linux, distribuciones

- Las más típicas:
  - Red Hat: Soporte técnico, servidores.
  - Mandrake: Sencillez por encima de todo.
  - SuSE: Seriedad alemana.
  - Debian: Creada por voluntari@s, Libertad.
    - Knoppix: Autoarrancable, Live-CD.
    - Ubuntu: Debian fácil y asequible.
  - Gentoo: Basada en la compilación de todos los programas, optimización.

GNU/Linux, distribuciones

- Las distribuciones suelen diferenciarse en:
  - El modo de instalar el software: hay distribuciones que instalan el software de forma manual, otras utilizan sistemas de “paquetes” de software, que pueden instalarse de forma automática, etc.
  - La forma de configurar el sistema: hay distribuciones con multitud de asistentes, otras más técnicas.
  - El sector al que están dirigidas: hay distribuciones para el público en general y otras centradas en un único aspecto: la seguridad, la facilidad de uso, los juegos, etc.

¿Cómo conseguir GNU/Linux?

- Existen múltiples maneras de conseguir GNU/Linux: Como es software libre…
  - lo puedes copiar de un amigo sin problemas, ¡está permitido!
  - lo puedes descargar de Internet sin problemas, cuantas veces quieras y para lo que quieras.
  - hay empresas que lo empaquetan y preconfiguran para facilitarnos las cosas y nos lo venden (Red Hat, Mandrake, SuSE, etc.).
  - hay organizaciones que lo regalan (Ubuntu, Junta de Extremadura, etc).
  - muchas revistas incluyen versiones completas en sus CDs o DVDs, por poco dinero.

¿Cómo conseguir GNU/Linux?

- Sitios de Internet donde descargar GNU/Linux:
  - http://www.linuxiso.org
  - http://www.linuxhelp.net/isos/
- Ubuntu regala CDs de GNU/Linux y te los envía gratuitamente a casa:
  - http://shipit.ubuntulinux.org

Ubuntu

- Distribución GNU/Linux basada en Debian
  - Impulsada por la Ubuntu Fundation creada por Mark Shuttleworth (Canonical Ltd.)
  - Su objetivo es llegar a cualquier tipo de usuario, en todo el mundo.
  - Live CD.
  - Sencilla de instalar
  - Tiene un buen soporte
  - www.ubuntu.com

Instalación de Ubuntu

- Ubuntu Dapper es un Live CD instalable
  - Es posible probarla sin instalar nada, y se ejecuta desde la memoria RAM.
  - Podemos iniciar la instalación desde el icono que hay en el escritorio
  - Es muy sencilla de instalar. Hay dos pasos críticos:
    - Particionado: Los sistemas GNU/Linux necesitan al menos dos particiones. Una para datos y otra para swap. Hay que hacerlo con cuidado si existen datos en el disco.
    - Instalación del gestor de arranque.

Instalación de Ubuntu

- Gestor de arranque
  - Se situa en el sector de arranque del disco. La máquina lo lee e inicia el sistema operativo.
  - Ubuntu instala GRUB
    - Es posible iniciar otros sistemas operativos desde él además de GNU/Linux. Podemos tener varios S.O instalados.
    - Durante la instalación se detectan otros S.O y se permite configurar el arranque para ellos.
    - Cada vez que iniciemos la máquina podemos elegir el S.O que queremos usar si tenemos varios.

Entorno Gráfico

- El entorno gráfico en GNU/Linux no es un todo inseparable del resto del S.O. Es una capa más.
- Podemos, por lo tanto, cambiarlo “fácilmente” o incluso quitarlo si no lo necesitamos (servidores).
  - Se puede dividir en dos partes
    - X Window System, librerías básicas para pintar cosas en pantalla (X.Org, XGL, XFree86).
    - El gestor de ventanas/entorno de escritorio. Proporciona el aspecto visual y la interacción más compleja con el usuario.

Entorno Gráfico

- X Window System
  - Posee una arquitectura cliente-servidor.
    - Es posible ejecutar aplicaciones en una máquina remota y ver el resultado en la nuestra.
      - Servidor X: sirve el dispositivo gráfico. Es el que pinta en la pantalla.
      - Clientes: los programas que usan el servidor X gráfico.
  - X.Org es el más usado actualmente.
    - Creado a partir de XFree86 debido la problemas de licencia
    - Ha itnroducido novedades y sobre todo cambios en la mantenibilidad del cógido fuente.

Entorno Gráfico

- Entorno de Escritorio/Gestor de Ventanas:
  - Proporcionan un “Look & Feel” para las aplicaciones
  - ¡Para gustos, los colores! Cada usuario puede decidir aquel que más se ajusta a sus necesidades o gustos personales.
    - GNOME: Por defecto en Ubuntu
    - KDE: Mucho mejor que GNOME, :D. KDE Powa!
    - Xfce: Más ligero que los anteriores. Minimalista
    - WindowMaker, FluxBox, Ion3, ...

Gestor de Paquetes – Primera Parte

- Las distros Debian-like poseen un gestor de paquetes de software muy potente
  - Existen repositorios de software empaquetado y preconfigurado listo para usar.
  - El gestor se encarga de:
    - buscar en los repositorios
    - determinar las dependencias (librerías, otros programas, etc.)
    - descargar todo
    - instalarlo de manera automática

Gestor de Paquetes – Primera Parte

- Permite también actualizar el software instalado de forma sencilla:
  - Ya, ¿pero esto no lo hacen ya “otros” S.O?
    - No exactamente...
  - El gestor de paquetes de Ubuntu-Debian permite actualizar TODO el software instalado en el sistema mediante de paquetes. :-O
    - S.O (núcleo, librerías básicas, programas básicos...)
    - Pero además, navegadores, correo, oficina, grabación CD-DVD, servidores, juegos, ... TODO.

Gestor de Paquetes - Primera Parte

- Instalación de nuevos programas: “Synaptic”
- Actualización de paquetes: “Update Manager”
  - Existen dos tipos de actualizaciones:
    - Arreglo de bugs y versiones nuevas. Se producen a menudo, casi diariamente.
    - Actualizaciones de versión de la distribución
      - En Ubuntu se producen cada 6 meses.
      - Introducen cambios importantes en todo el sistema.
      - Actualmente la versión es Ubuntu Dapper.

Programas Equivalentes

- Internet / Red
  - Navegadores
    - Gráficos: firefox, konqueror
    - Consola: lynx, Elinks
  - Correo:
    - Gráficos: evolution, kmail
    - Consola: mutt
  - Mensajería instantánea:
    - Gráficos: gaim, kopete

Programas Equivalentes

- Internet
  - IRC
    - Gráficos: bitchx, ksirc
  - P2P:
    - amule, mldonkey, bittorrent
- Editores de texto
  - Gráficos: gedit, kate
  - Consola: vi,vim, nano, joe, pico

Programas Equivalentes

- Multimedia
  - Reproductores de vídeo y DVD
    - xine, mplayer, ogle
  - Grabación de CD y DVD
    - k3b,
  - Reproductores de música
    - xmms
    - amarok (el mejor reproductor del mundo) XD

Programas Equivalentes

- Oficina
  - Suite OpenOffice.Org
  - Suite KOffice
- Desarrollo/ Programación
  - IDE
    - anjuta, kdevelop, monodevelop, eclipse
  - Compiladores
    - gcc, g++, python, java, pascal, haskell, ...

Programas Equivalentes

- Gráficos
  - Editores 2D
    - gimp
  - Visores PDF: kpdf, xpdf
- Editores 3D
  - blender
- Y juegos ...
- Y existen miles de ellos más en los repositorios.

GCC

- GNU C Compiler
  - Es el compilador básico del sistema
    - Permite compilar C y C++
    - Todo el núcleo y los programas básicos del S.O han sido compilados con él.
    - Aunque se llama compilador realiza también los procesos de ensamblado y linkado.
  - Depurador: gdb

GCC

- Ejemplo típico

```
#include <stdio.h>


void main(void)
{
        printf("Hello World!\n");
}
```

$gcc helloworld.c **-o** helloworld

GCC

- El gcc admite muchas opciones
- Más comunes
  - Directorios de includes: -Iincludedir
  - Librerías con las que linkar: -Llibname
  - Optimizaciones
  - Arquitectura
  - man gcc

Make

- Utilidad para mantener grandes programas
  - Para evitar que cada vez que hacemos un cambio sea necesario recompilar todos los ficheros fuente del proyecto.
  - Esta utilidad determina que ficheros necesitan ser recompilados y los construye de nuevo
  - Ademas construye los ficheros en el orden determinado (dependencias entre partes del programa)

Make

Estructura principal

```
target: {dependencias}
        comando
```

```
main: main.c
        gcc main.c -o main
```

target: etiqueta que identifica el bloque

dependencias: otros targets que deben ejecutarse antes

comando: que debe hacerse cuando se ejecute el target

make -f Makefile {target}

Make

main.c

```
#include <stdio.h>

int main(void)
{
        printf("Resultado: %i\n",suma(2,4));
}
```

suma.c

```
int suma(int a, int b)
{
        return a + b;
}
```

Makefile

```
all: suma

suma: suma.o main.o
        gcc suma.o main.o -o suma

main.o: main.c
        gcc -c main.c -o main.o

suma.o: suma.c
        gcc -c suma.c -o suma.o

clean:
        rm -rf *o suma
```

Compilar un programa

- A veces necesitaremos compilar un programa a partir del código fuente del mismo
  - Porque no existen paquetes del mismo compilados en los repositorios
  - Porque queremos la última versión del mismo
  - Porque queremos optimizarlo para nuestra arquitectura
  - Porque queremos cambiar el código
    - Añadir un parche de terceros
    - O modificar nosotros mismo el código por alguna razón

Compilar un programa

- Casi todos el código fuente disponible para GNU/Linux suele seguir la técnica “configure && make && make install”
  - 1 - Ejecutar el script ./configure en el directorio del código fuente del programa. Esto prepara todo para la compilación y comprueba que esten presentes las librerías necesarias.
    - Necesitamos la versión de desarrollo de las librerías (*-dev). Suelen estar en los repositorios oficiales por lo que se pueden instalar con apt-get

Compilar un programa

- 2 - Ejecutar make. Lee el makefile y realiza toda la compilación de forma automática
- 3 - Ejecutar make install (root) para instalar el programa en el sistema.
  - El problema de esta instalación es que no esta controlada por el gestor de paquetes por lo que no se mantiene automáticamente (actualizaciones, desistalaciones, dependencias).
- Aunque este es el procedimiento más comun siempre hay que leer las instrucciones de compilación.

Entornos de Desarrollo

- Existen muchos entornos de desarrollo libres para GNU/Linux
  - Algunos ejemplos
    - Anjuta + Glade (C y C++)
    - Kdevelop (C, C++, Ruby, PHP, Python, Perl)
    - Monodevelop (C#)
    - Eclipse (Java, C, C++)

Introducción a GNU/Linux

Unai Aguilera

gkalgan @ gmail .com

Jon Valdés

juanval @ gmail . com

Créditos:

Pablo Garaizar Sagarminaga

Cursillos de Julio 2006