

Gnu/Hurd & Gnome

Gnu/Hurd & Gnome

Tabla de contenidos

1. Introducción	1
1.1. El proyecto en cuestión	1
1.2. Motivación	1
1.3. Los problemas de este proyecto	2
2. Sistemas Operativos.....	3
2.1. Características básicas de un Sistema Operativo	3
2.2. Sistema de Ficheros	3
2.2.1. FAT 32	4
2.2.2. NTFS	4
2.2.3. ext2	4
2.2.4. ext3	4
2.3. Multiproceso	4
2.4. Concurrencia	4
2.5. Paginación	5
2.6. Multiusuario	5
3. El núcleo del sistema operativo	6
3.1.	6
4. El Micronúcleo de un sistema operativo.....	7
4.1.	7
5. El Proyecto GNU.....	8
5.1.	8
6. Comandos de Hurd.....	9
6.1. Grub	9
6.2. Montando dispositivos	10
6.3. Configurando la red.....	11

Capítulo 1. Introducción

Este proyecto que teneis en las manos ha sido generado para ser presentado en el *I Concurso Universitario de Software Libre*. en este concurso se podían presentar el proyecto que a uno lo viniera en cara siempre y cuando, el código generado fuera libre, donde libre significa:

- * poder ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.)
- * poder estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente)
- * poder copiar el programa de manera que se pueda ayudar al vecino o a cualquiera
- * poder mejorar el programa, y hacer públicas las mejoras, de forma que se beneficie toda la comunidad.

Así pues este proyecto está debajo estas condiciones, que como veis no son nada restrictivas, si no todo lo contrario, cosa que cuando uno lo lee por primera vez no puede más que sorprenderse.

1.1. El proyecto en cuestión

Este proyecto consiste en una documentación (la que estáis leyendo) que describe primero como funciona un Sistema Operativo y cuáles son sus funciones, con esta base sentada, ya se puede describir qué es y qué hace un kernel, una vez analizado el kernel ya pasamos al Microkernel, para luego detallar el gestor de Microkernel Hurd. Una vez todo esto explicado de pe a pa ya sólo queda saber distinguir bien entre lo que es el kernel y lo que es el Sistema Operativo, que en nuestro caso es GNU. I finalmente, con todo esto asimilado ya podemos describir la interfície gráfica y sus funciones, en nuestro caso trataremos con Gnome.

Además de toda esta documentación, el objeto principal de este proyecto no es más que un CD de instalación que nos permita instalar en nuestro ordenador personal el sistema Debian GNU/Hurd junto con la interfície gráfica Gnome.

1.2. Motivación

La motivación principal de este proyecto es por mero amor al arte, también por mi interés en saber cómo funciona un ordenador por dentro y de cómo puede hacer tantas funciones, tener tantos programas ejecutandose a la vez sin prácticamente notar nada (todo tiene sus limites ;)).

También añadir que uno empieza a estar cansado de estudiar siempre lo que le dicen y que empieza a ser hora de ponerse a investigar en lo que a uno realmente le interesa, sin encontrarse con favoritismos/juego sucio empresarial y de demostrar, que como el tener acceso libre al código, uno sin tener mucha experiencia en el campo, y con la ayuda inagotable de Internet, puede aprender y desarrollar cualquier cosa, sin límite alguno.

1.3. Los problemas de este proyecto

Antes de empezar, es bueno analizar la situación actual en lo que se refiere a GNU/Hurd. Los proyectos GNU y Hurd son dos proyectos debajo de las bases del Software libre, por lo tanto cualquiera que quiera aportar algo en este proyecto es libre de hacerlo, uno puede colaborar de muchas formas, desde provando los programas (es decir, instalar el sistema GNU/Hurd en su ordenador y probarlo), hasta hacer el diseño de sus respectivas páginas web e incluso si uno se esfuerza puede actualizar parte del código del proyecto o incluso puede añadirle nuevas funciones, para poder empezar a hacer esto lo recomendable es primero agregarse a sus respectivas listas de correo, donde rápidamente verá por donde van los tiros y se podrá adaptar y actuar en consecuencia.

Capítulo 2. Sistemas Operativos

Como el propósito de este proyecto es crear un cd de instalación de un sistema operativo, primero debemos saber cuáles son las funciones de un Sistema Operativo

2.1. Características básicas de un Sistema Operativo

La función principal de un sistema operativo es la de hacer de interfície entre el usuario/cliente (puede ser otra màquina) y el ordenador (hardware). Esta interfície, Sistema Operativo, nos debe permitir utilizar los recursos del ordenador, ya sea su unidad de procesamiento, el disco duro o todos los periféricos que nos podamos imaginar.

Entrando en más detalle, el sistema operativo es el responsable de guardarnos correctamente nuestros datos en el disco duro (sistema de ficheros, Filesystem), debe permitir que un usuario pueda ejecutar más de una aplicación a la vez (multiproceso), para cada proceso distinto debe asignarle unos recursos concretos y limitados según los recursos disponibles y el número de procesos que se estan ejecutandose en ese momento en concreto, también debe permitir que todas las aplicaciones tengan acceso a todo los recursos, pero evitando cualquier conflicto (conurrencia), debe gestionar la rápida pero pequeña memoria RAM para agilizar los accesos a los datos (paginación) e incluso debe permitir que en un mismo ordenador se conecten más de un usuario a la vez y que no por ello los datos de uno se mezclen con los de otros y además debemos proteger los datos de unos y otros (multiusuario).

Todas estas funciones, son las funciones más relevantes que debe hacer un sistema operativo, encima, todo esto debe hacerse de la forma más rápida posible, pues el usuario quiere que cuando teclee una letra, esta aparezca de inmediato por pantalla, si se proyecta un video tambien es necesario que este se mueva a los 24 frames por segundo como mínimo.

Por suerte los ordenadores son muy rápidos y pueden hacer más de 3000 instrucciones por segundo, donde, por ejemplo, una suma de dos enteros tarda 7 instrucciones.

Veremos estas características con un poco de detalle, aunque si el lector (o sea, tu), está interesado en profundizar sobre el tema se puede coger el libro de Richard Stallings que además és ideal para la gente que no está muy metida en el tema.

2.2. Sistema de Ficheros

El sistema de ficheros es el que determina como guardar los datos en un dispositivo de memoria (disquete, disco duro, memoria flash), así pues depende de como organizamos los datos en el dispositivo

podremos acceder a ellos más rápido o más lento. Entender esto es trivial si nos imaginamos un cajón donde tenemos todo de objetos: lapices, gomas, reglas, ... si estos presentan cierto orden, es muy fácil encontrar lo que queremos, en cambio si esta todo apilado tendremos que remover los objetos para encontrar lo que estamos buscando. Pues con un disco duro, más o menos /igual.

De sistemas de ficheros hay muchos, los más conocidos: FAT32, ext2, ext3, ReiserFS y NTFS. Todos ellos con sus pros y sus contras. veremos éstos de manera rápida y a modo de introducción.

Bien, con esto ya solo queda añadir que en un mismo disco pueden haber distintos Sistemas de Ficheros, pero esto sí, cada uno de ellos tiene su espacio fijado, estos espacios son disjuntos, es decir, no puede ser que un mismo bloque de disco duro pertenezca a más de una partición de un sistema de ficheros a la vez (esto es trivial, dado que el caos que podría generar). Aunque con un sólo sistema de ficheros un sistema operativo tiene suficiente, es importante que esté preparado para gestionar cuántos más mejor, pues nos permitirá acceder a cualquier dispositivo magnético sin problemas y también nos permitirá escoger el sistema de ficheros que creamos que será mejor para la función que deberá realizar el ordenador.

2.2.1. FAT 32

2.2.2. NTFS

2.2.3. ext2

2.2.4. ext3

2.3. Multiproceso

Es necesario que un sistema operativo pueda administrar y gestinar más de un proceso a la vez, si no, no podríamos ni tant sólo ver lo que estamos tecleando por teclando, pues para esto es necesario un proceso que escupa por pantalla lo que s'está escribiendo y lo retenga a cada frame, además es necesario otro proceso que esté pendiente del teclado, para ver qué se teclea(aunque este proceso se haga por interrupciones, es necesario poder gestionar multiprocesos).

2.4. Concurrencia

No podemos dejar que dos procesos quieran utilizar un mismo recurso a la vez, pero también debemos asegurarnos que todos los procesos tienen acceso a todos los dispositivos (a no ser que el administrador o responsable del ordenador quiera impedir el uso de cierto dispositivo, por ejemplo impedir que se imprima desde cierto ordenador). Todo esto implica poner un orden y unas restricciones a cada uno de los procesos.

2.5. Paginación

Aunque también podemos encontrar otro método para gestionar la memoria: la segmentación, Linux y Hurd usan la paginación, es por eso que sólo me centraré en la paginación (aunque la segmentación es lo más o menos lo mismo sólo que el tamaño de las páginas es variable y por lo tanto se deben hacer unos cálculos distintos).

2.6. Multiusuario

Debemos permitir que muchos usuarios puedan acceder a un mismo ordenador (y dependiendo de los propósitos, que sólo un usuario pueda acceder al ordenador), esto nos obliga a impedir que un usuario pueda acceder a datos que no le conciernen (de otro usuario o del propio sistema) para evitar que haga alguna acción no deseada o que adquiera información privilegiada.

de aquí la identificación que nos pide el sistema nada más arrancar.

Visto todo esto ya tenemos una idea más clara de las funciones que un sistema operativo debe hacer, pero aún así nos falta saber una cosa, ¿quién hace toda esta tarea? la respuesta está en los dos siguientes capítulos.

Capítulo 3. El núcleo del sistema operativo

Todas las funciones descritas en el capítulo anterior se deben realizar lo más rápido posible, por eso, es muy importante saber gestionarlas bien. Esto nos determinará el rendimiento que obtengamos de la máquina(ordenador) en cuestión.

La forma de gestionar todas las funciones de un sistema operativo es la de crear un programa en un nivel lo más cercano al hardware posible que se encargue de toda la gestión de susodichas funciones. Así es como funciona Linux.

3.1.

Capítulo 4. El Micronúcleo de un sistema operativo

Como las llamadas a sistema implican cambios de `¿**"perfil"**?¿`, éstos son lentos, y como encima se tienen que hacer constantemente, su rendimiento se ve altamente afectado. Una solución es aligerar el núcleo, de esta manera la gran mayoría de las llamadas a sistema se harán en modo usuario, con lo que no tendremos que cambiar de `¿**perfil**?¿`. Esto parece muy bonito, pero genera nuevos problemas que antes no teníamos: ¿qué acciones paso a modo de usuario y cuáles no? ¿es igual de seguro el sistema?.

4.1.

Capítulo 5. El Proyecto GNU

El proyecto GNU fue iniciado por Richard Stallman, que viendo como día a día toda la industria del software optaba por privatizar los códigos de los programas, se propuso hacer un gran proyecto: Un sistema Operativo totalmente libre.

Una de las innovaciones de GNU es que en su propio sistema hay un manual de todas las aplicaciones que hay instaladas, esto se agradece y mucho, pues ya no dependemos de ningún manual con tropocientas páginas y rápidamente podemos descubrir como funciona cierto programa y cual es su modo de uso.

5.1.

Capítulo 6. Comandos de Hurd

A partir de aquí esta documentación deja su carácter teórico y pasa a un carácter práctico, para realizar esta documentación se ha usado especialmente estos dos enlaces: www.Google.com y http://www.gnu.org/software/hurd/users-guide/using_gnuhurd.txt

6.1. Grub

La función de Grub es la de poder permitir arrancar más de un sistema operativo en un mismo ordenador, por eso se encarga del arranque, y si está instalado correctamente, cuando iniciemos nuestro ordenador, Grub nos preguntará qué sistema queremos arrancar. Como GNU/Hurd es un sistema que está en fase de pruebas esto nos irá de maravilla porque así podremos tener otros sistemas más estables y con más recursos instalados (cualquier distribución de GNU/Linux nos servirá).

Desafortunadamente la configuración de Grub para que nos arranque nuestro GNU/Hurd es un tanto peculiar, para ello deberemos editar el archivo `/boot/grub/menu.lst` e añadirle las siguientes líneas:

```
title Debian GNU/Hurd root (hd1,0) kernel /boot/gnumach.gz root=device:hd1s1 module
(hd1,0)/hurd/ext2fs.static --multiboot-command-line=${kernel-command-line}
--host-priv-port=${host-port} --device-master-port=${device-port} --exec-server-task=${exec-task}
-T typed ${root} $(task-create) $(task-resume) module (hd1,0)/lib/ld.so.1 /hurd/exec
$(exec-task=task-create) savedefault boot # This entry automatically added by the Debian installer
for a non-linux OS # on /dev/hdb1 title Debian GNU/Hurd (recovery mode) root (hd1,0) kernel
/boot/gnumach.gz root=device:hd1s1 -s module (hd1,0)/hurd/ext2fs.static
--multiboot-command-line=${kernel-command-line} --host-priv-port=${host-port}
--device-master-port=${device-port} --exec-server-task=${exec-task} -T typed ${root}
$(task-create) $(task-resume) module (hd1,0)/lib/ld.so.1 /hurd/exec $(exec-task=task-create)
savedefault boot
```

Pero no deberemos poner siempre los mismos valores. Todo depende de la partición y de en qué disco hayamos instalado GNU/Hurd. Así que antes de editar el archivo `menu.lst`, será mejor iniciar nuestro ordenador con Grub y cuando estemos en el menú para seleccionar el sistema operativo pulsamos **c**, aquí vemos que estamos en una línea de comandos, en ella si escribimos:

```
grub> find /boot/gnumach.gz
```

```
hd(1,0)
```

Ésta es la forma que tiene Grub para identificar los discos duros, el primer número indica qué disco es al que queremos acceder, 0 indica que es el master del IDE primario y 3 indica que es el slave del IDE

secundario. El segundo valor indica en qué partición queremos acceder, así 0 indica que queremos acceder a la primera partición i M indica que queremos acceder a la partición m-ésima-1.

Pero luego tenemos la forma en que identifica los discos GNU/Hurd, que es distinta incluso de Linux, así hd0s1 indica que queremos acceder a la primera partición del disco master del IDE primario, en cambio hd3s2 indicaría que queremos acceder a la segunda partición del disco slave del IDE secundario. Nótese que las particiones empiezan desde la 1 hasta la N, en cambio en Grub empiezan desde la 0 hasta la N-1.

El recovery mode, es para acceder al sistema sin que éste reconozca ningún usuario, esto va bien cuando se pierde el password de root, o en GNU/Hurd para poder configurar el sistema por primera vez.

6.2. Montando dispositivos

Aunque el comando **mount** es reconocido por GNU/Hurd, éste sólo es reconocido por las entradas configuradas en el archivo `/etc/fstab` (en Linux igual, però si no está en `fstab` podemos utilizar igualmente `mount`, pasando más datos como parámetro). Así, en GNU/Hurd se usa el comando **settrans**. `Settrans` vendría a significar "activa traductor", los translators (traductores) seran quienes se encargaran de podernos comunicar con los dispositivos.

Pero estos dispositivos, antes los deberemos crear, si es que no estan creados, por esto es necesaria la instrucción `/dev/MAKEDEV` que nos creará los dispositivos que necesitemos, así por ejemplo si queremos acceder a la segunda partición del disco master del IDE primario (donde podríamos tener almacenado cualquier dato que nos interese), deberíamos ejecutar:

```
cd /dev
```

```
./MAKEDEV hd0s2
```

Un vez hecho esto podríamos editar con el archi `/etc/fstab` con **nano `/etc/fstab`** y añadir una línea similar a esta:

```
/dev/hd0s2 /home ext2 rw 0 2
```

```
mount /home
```

Con esto tenedríamos el home, de por ejemplo nuestro Linux anterior, que lo usáíamos también para GNU/Hurd (No olvidar que son compatibles).

Para montar disquetes haríamos lo siguiente:

```
settrans -a /media/floppy /hurd/ext2fs /dev/fd0
```

Para montar el lector de CD's que podríamos tener como master en el IDE secundario:

```
cd /dev
```

```
./MAKEDEV hd2
```

```
settrans -ac /media/cdrom /hurd/isofs /dev/hd2
```

Así en los directorios /media/floppy y /media/cdrom tendremos los contenidos del disquete y del cdrom respectivamente, e incluso en el disquete podemos escribir y copiar ficheros hasta allí Y veremos como realmente la disqueteera hace ruido escribiendo, cosa que es gratificante después de la ardua tarea de instalar el sistema base.

Un pequeño inciso es que no sé porque para poder acceder los disquetes es mejor darles formato con la siguiente instrucción:

```
mke2fs -b 4096 /dev/fd0
```

6.3. Configurando la red

Para poder configurar la tarjeta de red también haremos uso de la instrucción settrans.

```
# settrans -fgap /servers/socket/2 /hurd/pfinet -i eth0 -a a.b.c.d -g e.f.g.h -m i.j.k.l
```

Un poco de descripción de este comando: 'fg' fuerza a todos "translator" (traductor: quien nos permite comunicar con los dispositivos) a desactivarse. 'ap' activa los traductores (translators, no sé si usar la traducción o no), y los vuelve pasivos, así, al activarlos se descubre si existe algun error (stderr), y al volverlos pasivos guardamos la configuración, de modo que al reiniciar se volverá a activar dicho dispositivo.

La opción '-i' no indica la interficie que vamos a usar, esto la tarjeta de red. '-a' corresponde a la dirección IP, '-g' corresponde al gateway (dirección del router) y '-m' a la máscara de red (normalmente 255.255.255.0).