Gestión de fallos de energía en Linux

por Alejandro Martínez Fernández

17 de mayo de 2000

Contents

1 Introducción

La gestión de fallos de energía es un aspecto que a menudo queda relegado a un segundo plano a la hora de instalar y administrar redes de ordenadores. La alta disponibilidad de conexiones a la red de suministro eléctrico en los edificios modernos, y su aceptable fiabilidad, hacen que estas cuestiones sólo se plantéen a la hora de usar un portátil o de administrar un servidor de una red de trabajo. Las principales razones para diseñar un modelo de gestión de energía son:

- incremento de la productividad
- mayor disponibilidad del sistema
- mejoras ecológicas y económicas
- salvaguarda frente a fallos de alimentación

Como veremos en el próximo capítulo, por medio del APM (Advanced Power Management) podemos mejorar la productividad y disponibilidade de nuestros sistemas, además de ahorrarnos el coste económico y ecológico de parte del despilfarro energético.

En el capítulo 3, explicaremos los métodos de recuperación frente a fallos de alimentación basados en sistemas UPS.

Por último, en el capítulo 4, estudiaremos una aplicación de lo expuesto, en el marco del proyecto EHAS.

2 APM

2.1 £Qué es APM?

APM es un mecanismo diseñado por Intel y Microsoft para gestionar el consumo energético del ordenador.

APM se basa en una o más capas de software que proporcionan un servicio de gestión de energía en aquellos ordenadores cuyo hardware lo soporte.

APM define una interfaz software entre el software de gestión de energía específico del hardware, que reside en la BIOS, y un driver gestor de energía del sistema operativo. Esta interfaz enmascara los detalles del hardware, permitiendo al software de niveles superiores utilizar APM sin tener conocimiento de la interfaz hardware.

APM extiende la vida de las baterías del sistema y por tanto incrementa la productividad y la disponibilidad del sistema. Por otra parte el ahorro energético presenta ventajas en una doble vertiente: ecológica y económica.

Los componentes del sistema APM son:

BIOS APM: es el software que proporciona la interfaz entre la placa madre y los dispositivos y componentes cuyo consumo es gestionable.

Interfaz APM: definición de un conjunto de operaciones que permiten la interacción entre el driver APM y la BIOS APM.

Driver APM: conecta con la BIOS APM y controla la política de gestión de energía. El driver APM se comunica con las aplicaciones APM.

Aplicaciones APM: monitorizan y/o controlan la gestión de energía.

Módulos controladores de dispositivos APM: proporcionan una interfaz software para otros dispositivos añadidos.

2.2 Comprobación de Compatibilidad de APM con Linux

Para que APM funcione en un portátil o en un ordenador de sobremesa, la ROM BIOS del sistema debe soportar el estándar APM. Además para que APM funcione con Linux, la ROM BIOS del sistema debe soportar la versión 1.0 ó superior del estándar APM, y debe permitir también conexiones en modo protegido 32-bits. Es preferible un sistema que soporte APM 1.1 ó 1.2, ya que estas versiones proporcionan características añadidas que el driver y las aplicaciones pueden aprovechar.

Puede obtener información sobre la versión de APM con el comando `dmesg` y en el fichero `/proc/apm`.

2.3 APM y Linux

El soporte de APM tiene dos partes: soporte en el kernel y soporte en zona de usuario.

2.3.1 Soporte de APM en el kernel

Es posible que el kernel que viene con su distribución no tenga soporte APM (aunque la mayoría de las distribuciones modernas sí que lo incorporan por defecto). Si es así, deberá recompilar un kernel.

Activando la opción del kernel "Advanced Power Management BIOS support"[1], aparecen todas las características relacionadas con APM. Algunas de estas opciones son autoexplicativas, otras en cambio requieren una aclaración que se facilita en la siguiente ayuda. Veamos las que incluye un kernel 2.2.14:

CONFIG_APM_DISABLE_BY_DEFAULT: esta opción desactiva APM en tiempo de arranque a no ser que se especifique `apm=on` como parámetro del kernel.

CONFIG_APM_IGNORE_USER_SUSPEND: está opción hace que se ignoren las peticiones `USER SUSPEND`.

CONFIG_APM_DO_ENABLE: activa las características APM en tiempo de arranque. Cuando se desactiva la BIOS APM no adopta ninguna política de ahorro energético como respuesta a las llamadas Idle a la CPU.

CONFIG_APM_CPU_IDLE: Habilita las llamadas APM Idle CPU / Busy CPU (CPU inactiva / ocupada) dentro del bucle idle del kernel. En algunas máquinas, esto puede proporcionar un mayor ahorro energético, como el debido a un descenso en la velocidad de reloj de la CPU cuando

[1] En los kernels 2.2.x la configuración de APM se encuentra dentro del capítulo General Setup.

el ordenador está inactivo. Estas llamadas idle se ejecutan después de que el bucle idle se haya ejecutado durante un determinado periodo de tiempo (por ejemplo 333 mS). En el algunas máquinas, esto

CONFIG_APM_DISPLAY_BLANK: Activa el apagado de consola por medio de APM. Algunos portátiles pueden apagar la retroiluminación del LCD cuando el salvapantallas de la consola virtual de Linux apaga la pantalla. Advierta que esto sólo se utiliza por el salvapantallas de la consola virtual, y no apagará la retroiluminación cuando se utilice es sistema X Window. Esto tampoco tiene nada que ver con los monitores con ahorro energético que cumplen la normativa VESA. Más aun, esta opción no funciona para todos los portátiles – puede no apagar la retroiluminación, o puede simplemente escribir un montón de errores en la consola, especialmente si está utilizando gpm.

CONFIG_APM_POWER_OFF: Activa la posibilidad de apagar el ordenador despues de que el kernel de Linux se detenga (`halt`). Necesitará software que pueda apagar el ordenador (por ejemplo una versión del comando `halt(8)` ("`man 8 halt`")). Versiones recientes del paquete sysvinit estám disponibles en `ftp://metalab.unc.edu/pub/Linux/system/daemons/init` incluyen soporte para ello ("`halt -p`" cierra Linux y apaga el ordenador, si se ejecuta desde el nivel 0 de ejecución). Como con otras opciones APM, esta opción puede no funcionar correctamente con algunas implementaciones BIOS.

CONFIG_APM_IGNORE_MULTIPLE_SUSPEND: Esta opción es necesaria en los IBM Thinkpad 560, pero debería funcionar con el resto de portátiles. Ignora los múltiples eventos suspend o standby que puede devolver la BIOS APM mientras que se esté procesando uno todavía. Sin esta opción el Thinkpad 560 tiene problemas con el daemon de nivel de usuario apmd, y con el paquetes PCMCIA pcmcia-cs.

CONFIG_APM_IGNORE_SUSPEND_BOUNCE: Esta opción es necesaria en los Dell inspirion 3200 y en otros, pero debería ser segura para el resto de portátiles. Cuando está activa, se ignora cualquier evento de suspend de sistema que ocurra en los tres primeros segundos de un resume. Sin esta opción el Inspiron se apagará solo unos segundos despues de abrir el portátil, haciendo que sea necesario pulsar el botón de encendido para hacer resume por segunda vez.

CONFIG_APM_RTC_IS_GMT: active esta opción si su RTC (Real Time Clock, también llamado reloj hardware) guarda la hora GMT (Greenwich Mean Time o Hora del Meridiano de Greenwich). Desactive esta opción su su RTC guarda la hora local. Se recomienda que guarde la hora GMT en su RTC, porque así no tiene que preocuparse por los cambios de hora para ahorro de luz. La única razón para no usar GMT en su RTC es que ejecute también un Sistema Operativo que no entienda GMT.

CONFIG_APM_ALLOW_INTS: Normalmente se desabilitan las interrupciones externas mientras se llama a la BIOS APM para disminuir los efectos de una implementación BIOS que tenga un comportamiento incorrecto. La BIOS debe reactivar las interrupciones si es necesario. Por desgracia, algunas BIOSes no lo hacen - especialmente las que vienen en muchos de los nuevos Thinkpads de IBM. Si experimenta cuelgues cuando hace suspend, pruebe a activar esta opción. En otro caso, no la active.

Como hemos comentado es posible que no todas las características funcionen en todos los portátiles.

2.3.2 Soporte de APM en zona de usuario.

`apmd` es un daemon de monitorizacion de APM, y trabaja en conjunción con el driver BIOS APM del kernel del sistema operativo. Permite ejecutar un comando (tipicamente un shell script) cuando el driver informa de determinados eventos, y guarda un registro, mediante `syslogd(8)`, de ciertos cambios del estado de alimentación del sistema. Cuando la energía disponible en la batería es muy baja, puede alertar a todos los usuarios del sistema, empleando diversos metodos.

Cuando el driver APM del kernel notifica al *daemon* una peticion pendiente de suspend o standby, apmd invoca el comando apropiado, registra el evento, hace `sync(2)` de los datos a disco, duerme un pequenyo periodo de tiempo para asegurarse de que todos los datos se vuelcan a disco, y luego avisa al driver APM para que continue su operacion. No obstante, en caso de suspend "critico" (que indica un apagado de emergencia) solo se ejecuta el ultimo paso (avisar al driver para que continue).

`apm` lee `/proc/apm`[2] y presenta la salida en un formato legible. Como básicamente se proporciona información sobre el estado de las baterías, este comando es especialmente útil en portátiles con una BIOS APM. `apm` también permite poner el ordenador en modo *standby* o *suspend*.

Otros comandos relacionados son: xapm(1), cardctl(8), hdparm(8), syslogd(8)...

3 UPS

Por UPS (*Uninterruptible Power System*), o SAI (Sistema de Alimentación Ininterrumpida), se entiende un sistema capaz de reaccionar a los fallos de suministro eléctrico, entregando a los equipos que alimenta cierta energía que fue almacenada previamente.

La gestión de UPS en Linux proporciona un mecanismo para apagar límpiamente el ordenador en caso de fallo del suministro eléctrico. La idea es aprovechar el margen de minutos que el sistema permanece encendido alimentado por las baterías de la UPS para detener, parcial o completamente, aquellos servicios del

[2] Este pseudodirectorio contiene información del kernel en ejecución relacionada con APM.

sistema que puedan deteriorarse conun apagado brusco. Para ello es necesario definir una interfaz hardware y software entre el ordenador y la UPS, y tener un software específico en la máquina Linux.

3.1 Clasificación de UPS

Repasemos algunos conceptos de UPS. Las UPSs caen dentro de dos categorías, que podríamos llamar "listas" y "simples". La diferencia entre ambas es la cantidad de información que se puede obtener de la UPS y la cantidad de control que se puede ejercer sobre ésta.

UPS Simple

- Se conecta al ordenador por el puerto serie.
- Usa las líneas de control del módem para comunicarse con el ordenador.
- Puede señalizar si hay un corte energético.
- Típicamente puede señalizar si la batería tiene poca carga.
- El ordenador suele poder ordenar a la UPS que se apague.

UPS Lista

- Se conecta al ordenador por el puerto serie.
- Se comunica con el ordenador por medio de transferencias de datos normales a través del puerto serie.
- Típicamente incluye algún tipo de lenguaje que el ordenador puede emplear para obtener diferentes informaciones de la UPS, fijar diversos parámetros operativos de la UPS, y controlar la UPS (tales como apagarla).

Normalmente las UPSs listas se pueden manejar en modo simple. Esto es muy útil porque algunas compañías que fabrican UPS listas sólo revelan sus protocolos de comunicaciones a personas con las que han firmado un acuerdo de confidencialidad.

3.2 Hardware

Se necesita un cable que conecte el puerto serie de la UPS con el puerto serie del ordenador. En caso de no tener dicho cable, no estar seguro de que funcione o no con `powerd`, no saber cómo funciona en absoluto, o necesitar uno nuevo que funcione, encontrará útil la información contenida en [1].

3.3 Software

Basicamente, todo lo que se necesita es un binario `powerd` que funcione, que habitualmente se encuentra en `/sbin/powerd` y forma parte del paquete SysVinit. Las versiones actuales de `init` incluyen los siguientes eventos:

- powerfail: anuncia un corte eléctrico
- powerfailnow: apaga el ordenador de manera inmediata
- powerok: aborta el apagado del ordenador

Como ya hemos comentado, el único problema que puede tener es que su cable puede no corresponderse con la configuración de `powerd`, en cuyo caso deberá recablearlo, o hacerse con una copia de `powerd.c` y modificarlo para que funcione con su cable.

Alternativamente puede utilizar otros programas que admiten diversas configuraciones y que aparecen listados en [1].

3.4 £Qué es necesario?

1. Conectar el ordenador a la UPS.
2. Conectar el puerto serie del ordenador a la UPS con un cable especial.
3. Ejecutar powerd (o algún programa equivalente) en el ordenador.
4. Configurar su init para que haga algo razonable cuando sucedan eventos powerfail y powerok (como iniciar el apagado (shutdown) y abortar (kill) los apagados (shutdowns) en ejecución, respectivamente, por ejemplo).

3.5 £Cómo se supone que funciona?

La tarea de la UPS: Cuando falla la alimentación, la UPS continúa alimentando al ordenador e indica que hay un corte de alimentación disparando un relé o activando un optoacoplador en el puerto de control.

La tarea del cable: El cable se diseña de tal forma que cuando la UPS dispara el citado relé, una línea de control particular del puerto serie (típicamente DCD) se activa a nivel alto.

La tarea de powerd: El *daemon* powerd monitoriza el puerto serie. Mantiene a nivel alto/bajo cualquier línea de control del puerto serie que la UPS necesite tener a nivel alto/bajo (típicamente, DTR debe estar a nivel alto y cualquier línea que apague la UPS debe mantenerse a nivel bajo).
*Cuando powerd detecta que la línea de control de la UPS pasa a nivel alto, inicia una conversación con init a través de un archivo FIFO, escribiendo un carácter cada vez que se detecta un cambio en el estado de la UPS. (Antiguamente, cuando `powerd` detectaba que la línea de control de la UPS pasaba a nivel alto/bajo, escribía FAIL/OK en `/etc/powerstatus` y mandaba la señal SIGPWR al proceso `init`).

La tarea de init (entre otras):

* `init` procesa la entrada del archivo FIFO y ejecuta la entrada del `/etc/inittab` correspondiente. (Antiguamente cuando recibía un SIGPWR, miraba en `/etc/powerstatus`. Si contenía FAIL ejecutaba la entrada powerfail de `/etc/inittab`. Si contenía OK ejecutaba la entrada powerokwait de `/etc/inittab`).

3.6 £Cómo configurarlo todo?

Edite `/etc/inittab`. Debería aparecer algo como:

```
# What to do when power fails/returns.
pf::powerfail:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
```

Edite `/etc/init.d/powerfail` (si no existe créelo). Y haga que contenga algo como:

```
#!/bin/bash
# Script powerfail para portátil
case "$1" in
    start)
        /sbin/shutdown -h +5 "Detectado fallo de alimentación. Parada en 5 minutos."
        ;;
    now)
        /sbin/shutdown -h now "Detectado fallo crítico de alimentación. Parada inmediat
        ;;
    stop)
        /sbin/shutdown -k
        ;;
    *)
        echo "Error en parámetro de powerfail"
esac
```

Modifique este script a su gusto (en la sección de este documento podrá ver un enfoque totalmente distinto). Advierta también que dependiendo de la versión de `shutdown` que esté utilizando, el contenido de este script puede ser tan sencillo como el que acabamos de mostrar o puede que necesite hacer tareas más complicadas; por ejemplo:

```
kill `ps -aux | grep "shutdown" | grep -v grep | awk '{print $2}'`
```

Haga que `init` vuelva a procesar el fichero inittab con el comando:

```
telinit q
```

Edite `/etc/rc.d/rc.local` (o el script correspondiente en su distribución) para que `powerd` se ejecute en el arranque. La sintáxis es:

```
powerd <line>
```

Sustituya `<line>` con el puerto serie que conecta con la UPS, tal como `/dev/ttyS1`.

3.7 Comprobando el funcionamiento

Conecte el puerto serie al puerto serie de la UPS. *NO CONECTE EL ORDENADOR A LA UPS TODAVÍA.* Alimente la UPS, encienda la UPS y la alimentación y ejecute powerd.

1. Desconecte el enchufe de la UPS.
   - Compruebe que la luz sigue encendida.
   - Compruebe que se ejecuta `/etc/init.d/powerfail start`.
   - Compruebe que se ejecuta `shutdown`.

2. Vuelva a conectar el enchufe de la UPS.
   - Compruebe que la luz sigue encendida.
   - Compruebe que se ejecuta `/etc/init.d/powerfail stop`.
   - Compruebe que ya no se ejecuta `/etc/init.d/powerfail start`.
   - Compruebe que ya no se ejecuta shutdown.

3. Vuelva a desconectar el enchufe de la UPS. Déjelo desconectado y compruebe que el ordenador se apaga correctamente pasado el periodo especificado de tiempo (en nuestro ejemplo 5 minutos).

4. La Parte Peligrosa. Después de que todo parezca que funciona apropiadamente, apague el ordenador y enchúfelo a la UPS.
   - Ejecute un script que haga sync del disco duro cada segundo más o menos.
   - Simultáneamente ejecute un segundo script que haga find sobre todo el disco duro.[3]
   - Ahora, tire del enchufe de la UPS, compruebe otra vez que `shutdown` se está ejecutando y espere. Asegurese de que el ordenador se apaga limpiamente antes de que la batería de la UPS se agote. Esto es peligroso porque si la alimentación se va antes de que el ordenador se apague, puede acabar con un sistema de archivos corrupto, y quizás incluso pueda perder todos sus ficheros. Seguramente quiera hacer una copia de seguridad completa del sistema antes de ejecutar este test, y fijar un tiempo pequeño de shutdown para empezar.

ąFelicidades! Ahora tiene un ordenador Linux protegido por UPS que se apagará limpiamente cuando se corte la alimentación.

[3]Lo primero es para hacerlo un poco más seguro y lo segundo para ayudar a consumir bastante enegía.

3.8 Mejoras

Hasta aquí la teoría básica de UPS bajo Linux. Todo lo comentado ha sido adaptado de [1]. En el momento de escribir este documento, dicho howto se encuentra algo anticuado pero sigue siendo una buena fuente de ideas para todo lo relacionado con UPSs y Linux. Desde dicho documento se le anima a que trastée con `powerd.c` y con el script `powerfail`, y lo ajuste a sus necesidades.

El siguiente capítulo es una muestra de cómo el tema de la gestión de UPSs puede ser ampliado para recoger otros sistemas de alimentación, como el caso de un sistema con alimentación fotovoltaica.

4 SAID

La descripción detallada de los gateways del proyecto EHAS se encuentra en proceso de elaboración. A nivel práctico, y centrándonos en el tema de la alimentación, tenemos que los gateways están diseñados para funcionar a partir de energía solar fotovoltaica. Los elementos empleados son: células solares, transforman la enegía solar fotovoltaica en pequeñas corrientes eléctricas; regulador, protege la batería de sobrecargas y descargas excesivas desconectando los paneles o la carga, en uno u otro caso; batería(s), almacenan la energía eléctrica; fuente de alimentación DC/DC, se encarga de obtener las tensiones necesarias para el resto del sistema a partir de los niveles de salida del regulador.

4.1 Construyendo un cable regulador-ordenador

Aunque nuestro sistema de alimentación presenta cierta similitud con las UPSs, ya que proporciona un sistema de alimentación ininterrumpida, no incluye los mecanismos de comunicación necesarios para apagar límpiamente el ordenador cuando se descarguen las baterías.

El regulador que hemos empleado incluye un LED que señaliza el estado de la batería. Es necesario adaptar esta señal para que pueda ser tratada automáticamente por el ordenador.

En nuestro caso los estados del LED son:

- apagado: batería normal
- parpadeando: batería baja
- encendido: batería cortada

Tenemos que apagar el ordenador cuando detectemos que la batería está baja (LED parpadeando) y antes de que el regulador la desconecte (LED encendido).

Con un sencillo circuito integrador conseguimos generar una señal contínua a partir del parpadeo del LED. A la salida de este circuito tendremos 0 cuando la batería se encuentre por encima de cierto umbral (aproximadamente unos 11,2v) y 1 cuando se encuentre por debajo de él.

El siguiente paso es introducir esta señal en alguno de los puertos de comunicaciones del ordenador. Como nuestro ordenador está destinado a hacer de pasarela entre sistemas radio-teléfono, debemos reservar dos puertos serie para la TNC radio y el módem. Para no tener que recurrir a PCs con más puertos serie, decidimos aprovechar el puerto paralelo para introducir esta señal en el ordenador.

En la documentación que estamos elaborando sobre el gateway se podrán consultar los diagramas de señal, PCBs, y el resto de la especificación hardware del sistema.

4.2 El software

4.2.1 El monitor said

El programa said monitoriza el estado de la alimentación a través de la señal BUSY del puerto paralelo. El código completo de este programa se encontrará en breve disponible en `http://viracocha.gbt.tfo.upm.es`

4.2.2 A revueltas con los scripts de inicio[4]

Edite el fichero `/etc/init.d/powerfail` (si no existe, créelo):

```
#!/bin/bash
# Script powerfail para máquina gateway
case "$1" in
     start)
         /etc/init.d/rcpf pf
         ;;
     now)
         /sbin/shutdown -r now "Alimentación recuperada; El sistema se va a reiniciar"
         ;;
     *)
         echo "Opción no reconocida"
esac
```

En caso de que llegue la señal `powerfail` (en nuestro caso: batería baja), se ejecuta el script `powerfail` con el parámetro `start` que a su vez invoca al script `rcpf`. Lo elegante sería poder definir un nivel de ejecución "de usuario", y cambiar a él cuando llega la señal `powerfail`. El problema es que estas señales de aviso no definen nuevos niveles de ejecución, por lo que tenemos que conformarnos con simularlo a través de la ejecución de un script.

El archivo `/etc/init.d/rcpf` es una pequeña variación de `/etc/init.d/rc`. La función de `rc` (entre otras) es ejecutar una serie de scripts situados en los

[4]Los localización y función de los scripts de inicio varía según la distribución de Linux. En este punto nos centraremos en la configuración de una Debian. Advierta que otras distribuciones sitúan `/etc/init.d` en `/etc/rc.d/init.d`.

directorios `/etc/rcN.d`[5] al cambiar de nivel de ejecución. Como `powerfail` no cambia de nivel de ejecución, `rc` no ejecuta los scripts de parada `/etc/rcN.d/Knn*`. que son precisamente los que más nos interesan. Con `rcpf pf` ejecutamos todos los scripts contenidos en `/etc/rcpf.d` tanto los de parada `Knn*` como los de arranque `Snn*`.

Este nuevo pseudonivel de ejecución `pf` es muy parecido al nivel 0 en el que se cierra completamente el sistema. Para crearlo, primero copiamos todos los archivos de `/etc/rc0.d` a `/etc/rcpf.d` excepto `S20sendsigs`[6] y `S90halt`[7], y después añadimos uno propio `S99para` que se ejecutará en último lugar. El contenido de `S99para` es:

```
#!/bin/bash
sync
mount -n -o ro,remount /var
mount -n -o ro,remount /
```

Con este script lo que hacemos es volcar todos los datos a disco y remontar las particiones como sistemas de sólo-lectura; de tal manera que si se produce un corte de alimentación por parte del regulador, las particiones no queden corruptas.

El sistema es poco mantenible ya que este nuevo pseudonivel de ejecución `pf` no es estándar y por tanto los nuevos servicios que instalemos no insertarán sus scripts de parada en `/etc/rcpf.d`, siendo tarea del administrador copiarlos a mano.

4.2.3 Arrancando said automáticamente

En Debian basta con crear un fichero `/etc/rc.boot/0said` que contenga:

```
#!/bin/bash
# Arrancamos el monitor de batería
echo "Starting said."
/usr/local/sbin/said &
```

En Red Hat debemos añadir al final de `/etc/rc.d/rc.local` las siguientes líneas:

```
# Arrancamos el monitor de batería
echo "Starting said."
/usr/local/sbin/said &
```

[5] N es el nivel de ejecución (0-6)

[6] Si matamos todos los procesos también muere el monitor de alimentación `said`.

[7] No queremos apagar el ordenador (todavía).

References

[1] Advanced Power Management (APM) BIOS Interface Specification Rev.1.2
(c) 1996 Intel, Microsoft
ftp://ftp.microsoft.com/developr/drg/apm/apm12_ch.exe

[2] Battery Powered Linux Mini-HOWTO
Hanno Mueller, hanno@lava.de
http://www.lava.de/~hanno/

[1] The UPS Howto
Harvey J. Stein, hjstein@bfr.co.il, Berger Financial Research, Ltd.
http://www.linuxdoc.org/HOWTO/UPS-HOWTO.html

[4] Linux Ecology HOWTO
Werner Heuser <wehe@snafu.de>, Wade W. Hampton <whampton@staffnet.com>
http://www.linuxdoc.org/HOWTO/Ecology-HOWTO.html