El nucli

Josep Jorba Esteve

P07/M2003/02283

UOC
www.uoc.edu

Índex

Introducció 5

1. El nucli del sistema GNU/Linux 7

2. Personalitzar o actualitzar el nucli 15

3. Procés de configuració i compilació 18
3.1. Compilació nucli versions 2.4.x 18
3.2. Migració a nucli 2.6.x 24
3.3. Compilació de nucli versions 2.6.x 25
3.4. Compilació a Debian del nucli (Debian Way) 27

4. Apedaçar el nucli 29

5. Els mòduls del nucli 31

6. Futur del nucli i alternatives 33

7. Taller: configuració del nucli per a les necessitats de l'usuari 37
7.1. Actualitzar nucli a Debian 37
7.2. Actualitzar el nucli a Fedora / Red Hat 39
7.3. Personalitzar i instal·lar un nucli genèric 41

Activitats 43

Altres fonts de referncia i informació 43

Introducció

El nucli del sistema GNU/Linux (que habitualment s'anomena Linux) [Vasb] és el cor del sistema: s'encarrega d'arrencar el sistema i, una vegada aquest ja és utilitzable per les aplicacions i els usuaris, gestiona els recursos de la màquina en forma de gestió de la memòria, sistema de fitxers, entrada/sortida, processos i intercomunicació de processos.

El seu origen es remunta a l'agost de 1991, quan un estudiant finlandès anomenat Linus Torvalds va anunciar, en una llista de missatges (*news*), que havia creat el seu propi nucli de sistema operatiu que funcionava conjuntament amb programari del projecte GNU, i l'oferia a la comunitat de desenvolupadors perquè el provés i suggerís millores per a l'utilització. Així, es va constituir l'origen del nucli de l'operatiu, que més tard es diria Linux.

Una de les particularitats del Linux és que, seguint la filosofia del programari lliure, se'ns ofereix el codi font del propi sistema operatiu (del nucli), de manera que és una eina perfecta per a l'educació en temes de sistemes operatius.

Nota

El nucli Linux es remunta a l'any 1991, quan Linus Torvalds el va posar a disposició de la comunitat. És dels pocs operatius que, tot i ser molt utilitzat, permet disposar del seu codi font.

L'altre avantatge principal és que, en disposar de les fonts, les podem recompilar per a adaptar-les millor al nostre sistema, i podem configurar els seus paràmetres per a donar un rendiment millor al sistema.

En aquesta unitat veurem com es maneja aquest procés de preparació d'un nucli per al nostre sistema: com, partint de les fonts, podem obtenir una nova versió del nucli adaptada al nostre sistema. De la mateixa manera, veurem com es desenvolupa la configuració i la posterior compilació i com es realitzen proves amb el nou nucli obtingut.

1. El nucli del sistema GNU/Linux

El nucli o *kernel* és la part bàsica de qualsevol sistema operatiu [Tan87], en què descansa el codi dels serveis fonamentals per a controlar el sistema sencer. Bàsicament, la seva estructura es pot separar en una sèrie de components de gestió orientats al següent:

- Gestió de processos. Quines tasques s'executaran, en quin ordre i amb quina prioritat. Un aspecte important és la planificació de la CPU: com s'optimitza el temps de la CPU per a executar les tasques amb el major rendiment o interactivitat possible amb els usuaris?
- Intercomunicació de processos i sincronització. Com es comuniquen tasques entre elles, amb quins diferents mecanismes, i com poden sincronitzar-se grups de tasques?
- Gestió d'entrada/sortida (E/S). Control de perifèrics i gestió de recursos associats.
- Gestió de memòria. Optimització de l'ús de la memòria, sistema de paginació i memòria virtual.
- Gestió de fitxers. Com el sistema controla i organitza els fitxers presents en el sistema i l'accés.

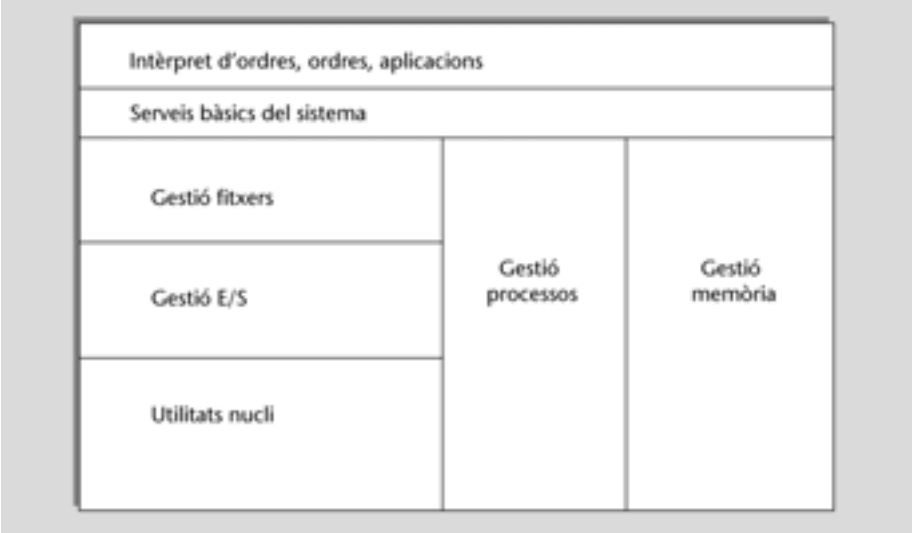

Figura 1. Funcions bàsiques d'un nucli respecte a les aplicacions i ordres executades

En els sistemes propietaris, el nucli està perfectament "ocult" sota les capes del programari del sistema operatiu, i l'usuari final no té una perspectiva clara de què és el nucli, i no té cap possibilitat de canviar-lo o optimitzar-lo, excepte per l'ús d'esotèrics editors de "registres" interns o programes especialitzats de tercers (normalment d'alt cost). A més, el nucli sol ser únic, és el que el fabricant proporciona, i aquest es reserva el dret d'introduir les modificacions que

vulgui i quan vulgui, i de tractar els errors que apareguin en terminis no estipulats, mitjançant actualitzacions que ens ofereixen com a "pedaços" d'errors.

Un dels principals problemes d'aquesta aproximació és, justament, la disponibilitat d'aquests pedaços, disposar de les actualitzacions dels errors al seu temps i, si són de seguretat, encara més, ja que fins que no estiguin corregits no podem garantir la seguretat del sistema per a problemes ja coneguts. Moltes organitzacions, grans empreses, governs, institucions científiques i militars no poden dependre dels capricis d'un fabricant per a solucionar els problemes de les aplicacions crítiques.

En aquest cas, el nucli Linux ofereix una solució de codi obert, amb els conseqüents permisos de modificació, correcció, possibilitat de generació de noves versions i actualitzacions de manera ràpida, per part de qualsevol que vulgui i tingui els coneixements adequats per realitzar-les.

Això permet als usuaris crítics controlar millor les seves aplicacions i el propi sistema, i poder muntar sistemes amb el propi operatiu "a la carta", personalitzat al gust de cada un. I disposar, al mateix temps, d'un operatiu amb codi obert, desenvolupat per una comunitat de programadors coordinats des d'Internet, accessible per a l'educació per disposar del codi font i abundant documentació, o per a la producció final dels sistemes GNU/Linux adaptats a necessitats individuals o d'un col·lectiu determinat.

En disposar de codi font, es poden aplicar millores i solucions de manera immediata, a diferència del que passa amb el programari propietari, en què hem d'esperar les actualitzacions del fabricant. A més, podem personalitzar el nucli tant com necessitem, requisit essencial, per exemple, en aplicacions d'alt rendiment, crítiques en el temps o solucions amb sistemes encastats (com dispositius mòbils).

Seguim amb una mica d'història (ràpida) del nucli [Kera] [Kerb]: el va començar a desenvolupar un estudiant finlandès anomenat Linus Torvalds, el 1991, amb la intenció de realitzar una versió semblant al Minix [Tan87] (versió per a PC del UNIX [Bac86]) per al processador 386 d'Intel. La primera versió publicada oficialment va ser la de Linux 1.0 el març de 1994, en la qual només s'incloïa l'execució per a l'arquitectura i386 i suportava màquines d'un sol processador. El Linux 1.2 es va publicat el març de 1995, i va ser la primera versió a donar cobertura a diferents arquitectures com Alpha, Sparc i Mips. El juny de 1996, el Linux 2.0 va afegir més arquitectures i va ser la primera versió a incorporar suport multiprocessador (SMP) [Tum]. El gener de 1999, en el Linux 2.2, es van incrementar les prestacions d'SMP de manera significativa, i es van afegir controladors per a una gran quantitat de maquinari. En la 2.4, el gener del 2001, es va millorar el suport SMP, es van incorporar noves arquitectures suportades i es van integrar controladors per a dispositius USB, PC card (PCMCIA dels portàtils), part de PnP (*plug and play*), suport de RAID i volums, etc. El desembre de 2003, en la branca 2.6 del nucli es millora sensiblement el

suport SMP, la resposta del sistema de planificació de la CPU, s'utilitzen en *threads* en el nucli, millora el suport d'arquitectures de 64 bits, hi ha suport de virtualització, i millora l'adaptació a dispositius mòbils.

Respecte al desenvolupament del nucli, des de la seva creació per Linus Torvalds el 1991 (versió 0.01), l'ha seguit mantenint ell mateix, però a mesura que el seu treball li ho permetia. I a mesura que el nucli madurava (i creixia), Linus ha mantingut les diferents versions estables del nucli ajudat per diferents col·laboradors, mentre continuava (en la mesura que podia) desenvolupant i recopilant aportacions per a l'última versió de desenvolupament del nucli. Els col·laboradors principals en aquestes versions han estat els següents [lkm]:

Nota

El nucli té els seus orígens en el sistema MINIX, desenvolupat per Andreu Tanenbaum, con un clon d'UNIX per a PC.

- 2.0 David Weinehall.
- 2.2 Alan Cox (també desenvolupa i publica pedaços per a la majoria de versions).
- 2.4 Marcelo Tosatti.
- 2.6 Andrew Morton / Linus Torvalds.

Per veure una mica la complexitat del nucli de Linux, vegem una taula amb una mica de la seva història resumida en les diferents versions i les seves mides del codi font. En la taula s'indiquen les versions de producció únicament; la mida (aproximada) està especificada en milers de línies (K) de codi font:

Versió	Data de la publicació	Línies (milers)
0.01	09–1991	10
1.0	03–1994	176
1.20	03–1995	311
2.0	06–1996	649
2.2	01–1999	1800
2.4	01–2001	3378
2.6	12–2003	5930

Com podem comprovar, hem passat d'unes deu mil línies a sis milions.

En aquests moments el desenvolupament continua en la branca 2.6.x del nucli, l'última versió estable, que inclou la majoria de distribucions com a versió per defecte (algunes encara inclouen algunes 2.4.x, però 2.6.x sol ser una opció en la instal·lació). Tanmateix, és imprescindible cert coneixement de les versions anteriors, ja que és fàcil trobar màquines amb distribucions antigues que no s'hagin actualitzat i que hàgim de mantenir, o en les quals hàgim de realitzar un procés de migració a versions més actuals.

Nota

Avui en dia, el nucli ha assolit un grau de maduresa i complexitat significatiu.

Durant el desenvolupament de la branca del 2.6, es van accelerar de manera significativa els treballs del nucli, ja que tant Linus Torvalds com Andrew Morton (que manté el Linux 2.6) es van incorporar (durant el 2003) a l'OSDL (Open Source

Nota

The Linux Foundation: www.linux-foundation.org

Development Laboratory) [OSDa], un consorci d'empreses amb la finalitat de promocionar l'ús *open source* i el GNU/Linux en l'empresa (en el consorci es troben, entre moltes altres empreses amb interessos en el GNU/Linux, HP, IBM, Sun, Intel, Fujitsu, Hitachi, Toshiba, Red Hat, Suse, Transmeta...). En aquells moments, es va produir una situació interessant, ja que el consorci OSDL va fer de patrocinador dels treballs, tant del mantenidor de la versió estable del nucli (Andrew) del desenvolupament (Linus), treballant a temps complet en les versions i en els temes relacionats. Linus es manté independent, treballant en el nucli, mentre Andrew se'n va anar a treballar a Google, on continuava els seus desenvolupaments a temps complet, realitzant pedaços amb diferents aportacions al nucli. Recentment, l'OSDL es va reconvertir en la fundació The Linux Foundation.

Cal tenir en compte que, amb les versions actuals del nucli, s'ha assolit un alt grau de desenvolupament i maduresa, i això farà que cada vegada s'ampliï més el temps entre la publicació de les versions (no, de les revisions parcials).

Un altre factor a considerar és el nombre de persones que treballen en el desenvolupament actual. Al principi hi havia unes quantes persones que tenien un coneixement global del nucli; avui en dia, en canvi, tenim moltes persones –prop de dues mil– que el desenvolupen amb diferents contribucions, encara que el nucli dur dels desenvolupadors s'estima en unes quantes dotzenes.

A més, hem tenir en compte que la majoria només tenen uns coneixements parcials del nucli, i ni tots treballen simultàniament ni la seva aportació és igual de rellevant (algunes d'elles només corregeixen errors senzills). De fet, són pocs (com els mantenidors) els que tenen un coneixement total del nucli. Això implica que es puguin allargar els desenvolupaments, i que les aportacions s'hagin de depurar per a evitar que entrin en conflicte entre elles, o decidir entre possibles alternatives de prestacions.

Respecte a la numeració de les versions del nucli del Linux ([lkm][DBo]), tenim en compte els aspectes següents:

a) Fins a la branca del nucli 2.6.x, les versions del nucli Linux es regien per una divisió en dues sèries: una era la denominada *experimental* (la numerada senar en la segona xifra, com 1.3.xx, 2.1.x o 2.5.x) i l'altra era la de producció (sèrie par, com 1.2.xx, 2.0.xx, 2.2.x, 2.4.x i més). La sèrie experimental consistia en versions que es movien ràpidament, i s'utilitzaven per provar noves prestacions, algoritmes o controladors de dispositiu, etc. Per la pròpia naturalesa dels nuclis experimentals, podien tenir comportaments impredictibles, com pèrdues de dades, bloquejos aleatoris de la màquina, etc. Per tant, no s'havien d'utilitzar en màquines destinades a la producció, tret que es volgués provar una característica determinada (amb els perills conseqüents).

Els nuclis de producció (sèrie parella) o estables eren nuclis amb un conjunt de prestacions ben definit, amb un nombre baix d'errors coneguts i controla-

dors de dispositius provats. Es publicaven amb menys freqüència que els experimentals, i hi havia varietat de versions, unes més bones que d'altres. Els sistemes GNU/Linux se solen basar en un determinat nucli estable escollit, que no és necessàriament l'últim nucli de producció publicat.

b) En la numeració actual del nucli Linux (utilitzada en la branca 2.6.x), es continuen conservant alguns aspectes bàsics: la versió està indicada per uns números *X.Y.Z*, en què normalment *X* és la versió principal, que representa els canvis importants del nucli; *Y* és la versió secundària, i normalment implica millores en les prestacions del nucli –*Y* és parell en els nuclis estables i senar en els desenvolupaments o proves–; i *Z* és la versió de construcció, que indica el número de la revisió d'*X.Y*, quant a pedaços o correccions fetes. Els distribuïdors no solen incloure l'última versió del nucli, sinó la que han provat amb més freqüència i poden verificar que és estable per al programari i components que ells inclouen. Partint d'aquest esquema de numeració clàssic (que es va seguir durant les branques 2.4.x, fins als inicis de la 2.6), va tenir algunes modificacions per a adaptar-se al fet que el nucli (branca 2.6.x) es torna més estable (fixant *X.Y* a 2.6), i cada vegada les revisions són menors (per significar un salt de versió dels primers números), però el desenvolupament és continu i frenètic.

En els últims esquemes, s'arriben a introduir quarts nombres, per a especificar canvis menors de *Z* o diferents possibilitats de la revisió (amb diferents pedaços afegits). La versió així definida amb quatre nombres és la que es considera estable (*stable*). També s'utilitzen altres esquemes per a les diverses versions de test (normalment no recomanables per a entorns de producció), com sufixos *-rc* (*release candidate*); els *-mm*, que són nuclis experimentals amb proves de diferents tècniques noves; o els *-git*, que són una mena de foto diària del desenvolupament del nucli. Aquests esquemes de numeració canvien constantment per adaptar-se a la manera de treballar de la comunitat del nucli i a les seves necessitats per a accelerar el desenvolupament.

c) Per obtenir l'últim nucli publicat, cal acudir al fitxer de nuclis Linux (a http://www.kernel.org) o al *mirror* local (a Espanya, http://www.es.kernel.org). També es podran trobar alguns pedaços en el nucli original que corregeixin errors detectats després de la publicació del nucli.

A continuació, destaquem algunes de les característiques tècniques ([DBo][Arc]) del nucli Linux:

- Nucli de tipus monolític. Bàsicament, és un gran programa creat com una unitat, però conceptualment dividit en diversos components lògics.

- Suport per a càrrega/baixada de porcions del nucli. És un suport a petició. Aquestes porcions s'anomenen *mòduls* i solen ser característiques del nucli o controladors de dispositiu.

- *Threads* de nucli. Per al funcionament intern s'utilitzen diversos fils (*threads*) d'execució interns del nucli, que poden estar associats a un programa d'usuari o bé a una funcionalitat interna del nucli. En el Linux, no es feia un ús intensiu d'aquest concepte. En les revisions de la branca 2.6.x es va oferir un suport millor, i gran part del nucli s'executa utilitzant diversos fils d'execució.

- Suport d'aplicacions *multithread*. És un suport d'aplicacions d'usuari de tipus *multithread*, ja que molts paradigmes de computació de tipus client/servidor necessiten servidors capaços d'atendre múltiples peticions simultànies amb un fil d'execució per cada petició o grup de peticions. El Linux té una biblioteca pròpia de *threads* que es pot utilitzar per a les aplicacions *multithread*, amb les millores que es van introduir en el nucli, que també han permès un ús millor per implementar biblioteques de fils (*threads*) per al desenvolupament d'aplicacions.

- Nucli no apropiatiu (*nonpreemptive*). Això implica que, dins del nucli, no es poden parar crides a sistema (en mode supervisor) mentre es resol la tasca de sistema, i quan aquesta acaba es reassumeix l'execució de la tasca anterior. Per tant, dins d'una crida, el nucli no pot ser interromput per atendre una altra tasca. Normalment, els nuclis apropiatius estan associats a sistemes que treballen en temps real, en què s'ha de permetre l'anterior per tractar esdeveniments crítics. Hi ha algunes versions especials del nucli del Linux per a temps real, que permeten això per mitjà de la introducció d'uns punts fixos en què es poden intercanviar. També s'ha millorat especialment aquest concepte en la branca 2.6.x del nucli, permetent, en alguns casos, interrompre algunes tasques reassumibles del nucli per tractar-ne d'altres, i resumir-ne l'execució posteriorment. Aquest concepte de *nucli apropiatiu* també pot ser útil per a millorar tasques interactives, ja que si es produeixen crides costoses per al sistema, poden provocar retards en les aplicacions interactives.

- Suport per a multiprocessador, el que s'anomena *multiprocessament simètric* (SMP). Aquest concepte acostuma a englobar màquines que van des del cas simple de 2 fins 64 CPU. Aquest tema s'ha posat d'especial actualitat amb les arquitectures de tipus *multicore*, i permet de dos a quatre o més *cores* de CPU en màquines accessibles als usuaris domèstics. El Linux pot utilitzar molts processadors, i cada un pot manejar una o més tasques. Però hi havia algunes parts del nucli que disminuïen el rendiment, ja que estaven pensades per a una única CPU i obliguen a parar el sistema sencer en determinats bloquejos. L'SMP és una de les tècniques més estudiades en la comunitat del nucli del Linux, i s'han obtingut millores importants en la branca 2.6. Ja, del rendiment SMP, depèn en gran manera l'adopció del Linux en els sistemes empresarials, en la faceta de sistema operatiu per a servidors.

- Sistemes de fitxers. El nucli té una bona arquitectura dels sistemes de fitxers, el treball intern es basa en una abstracció d'un sistema virtual (VFS o *virtual file system*), que es pot adaptar fàcilment a qualsevol sistema real. Com a resultat, el Linux és potser l'operatiu que més sistemes de fitxers suporta, des del propi *ext2*, fins a *msdos*, *vfat*, *ntfs*, sistemes amb diari (*journal*) com *ext3*, ReiserFS, JFS(IBM), XFS(Silicon), NTFS, *iso9660* (CD), *udf*, etc., i se'n van afegint més en les diferents revisions.

Vegem altres característiques menys tècniques (una mica de màrqueting):

a) El Linux és gratuït. Amb el programari GNU, i el que s'inclou en qualsevol distribució, podem tenir un sistema Unix complet, pràcticament, pel cost del maquinari, i els costos de distribució GNU/Linux són pràcticament nuls. Però no està de més pagar per una distribució completa, amb els manuals i suport tècnic, a un cost menor comparat amb el que es paga per alguns sistemes propietaris, o contribuir amb la compra al desenvolupament de les distribucions que més ens agradin, o ens siguin pràctiques.

b) El Linux és personalitzable. La llicència GPL ens permet llegir i modificar el codi font del nucli (sempre que tinguem els coneixements adequats).

c) El Linux s'executa en maquinari antic bastant limitat. És possible, per exemple, crear un servidor de xarxa amb un 386 amb 4 MB de RAM (hi ha distribucions especialitzades en baixos recursos).

d) El Linux és un sistema potent. L'objectiu principal en el Linux és l'eficiència. S'intenta aprofitar al màxim el maquinari disponible.

e) És d'alta qualitat. Els sistemes GNU/Linux són molt estables, sota proporció o ràtio d'errors, i redueixen el temps dedicat a mantenir els sistemes.

f) El nucli és bastant reduït i compacte. Es pot col·locar, amb alguns programes fonamentals, en un sol disc de 1.44 MB (hi ha diverses distribucions d'un sol disquet amb programes bàsics).

g) El Linux és compatible amb una gran part dels sistemes operatius. Pot llegir fitxers de gairebé qualsevol sistema de fitxers i comunicar-se per xarxa per oferir i rebre serveis de qualsevol d'aquests sistemes. A més, amb certes biblioteques, també pot executar programes d'altres sistemes (com MS-DOS, Windows, BSD, Xenix, etc.) en l'arquitectura x86.

h) El Linux rep un ampli suport. No hi ha cap altre sistema que tingui la rapidesa i quantitat de pedaços i actualitzacions que té el Linux, ni en els sistemes propietaris. Per a un problema determinat, hi ha infinitat de llistes de correu i fòrums, que en poques hores permeten solucionar qualsevol problema. L'únic problema és amb els controladors de maquinari recent, que molts fabricants

encara es resisteixen a proporcionar si no són per a sistemes propietaris. Però això canvia a poc a poc, i alguns dels fabricants més importants de sectors com targetes de vídeo (NVIDIA, ATI) i impressores (Epson, HP…) ja comencen a proporcionar els controladors per als seus dispositius.

2. Personalitzar o actualitzar el nucli

Com a usuaris o administradors de sistemes GNU/Linux, hem de tenir en compte les possibilitats que ens ofereix el nucli per adaptar-lo a les nostres necessitats i equips.

Normalment, construïm els nostres sistemes GNU/Linux a partir de la instal·lació, en els nostres equips, d'alguna de les distribucions del GNU/Linux, que poden ser comercials com Red Hat, Mandriva i Suse, o "comunitàries" com Debian i Fedora.

En el moment de la instal·lació, aquestes distribucions aporten una sèrie de nuclis Linux binaris ja preconfigurats i compilats, i normalment hem de decidir quin nucli del conjunt dels disponibles s'adapta millor al nostre maquinari. Hi ha nuclis genèrics, nuclis orientats a dispositius IDE o a SCSI, altres que ofereixen una barreja de controladors de dispositius [AR01], etc.

Una altra opció d'instal·lació acostuma a ser la versió del nucli. Normalment, les distribucions fan servir una instal·lació que consideren prou estable i provada com perquè no causi problemes als usuaris. Per exemple, en el dia d'avui, moltes distribucions vénen amb versions 2.6.x del nucli per defecte, ja que es considera la versió més estable del moment (en què va sortir la distribució). En alguns casos, en el moment de la instal·lació s'ofereix la possibilitat d'utilitzar, com a alternativa, versions més modernes amb un suport millor per a dispositius més moderns (d'última generació), però potser no tan provades en el moment en què es publica la distribució.

A més, els distribuïdors solen modificar el nucli per millorar el comportament de la seva distribució o corregir errors que han detectat en el nucli en el moment de les proves. Una altra tècnica bastant comuna en les comercials consisteix a deshabilitar prestacions problemàtiques, que poden provocar errors als usuaris o que necessiten una configuració especifica de la màquina. D'altres vegades, una determinada prestació no es considera prou estable per incloure-la activada.

Nota

La possibilitat d'actualitzar i personalitzar el nucli ofereix una bona adaptació a qualsevol sistema, i en permet l'optimització i la sintonització amb el sistema.

Això ens porta a considerar que, per més bé que un distribuïdor faci la feina d'adaptar el nucli a la seva distribució, sempre podem trobar una sèrie de problemes:

- El nucli no està actualitzat en l'última versió estable disponible; alguns dispositius moderns no reben suport.
- L'estàndard nucli no disposa de suport per als dispositius que tenim perquè no han estat habilitats.

- Els controladors que ens ofereix un fabricant necessiten una nova versió del nucli o modificacions.
- A la inversa, el nucli és massa modern, i tenim maquinari antic que ja no té suport en els nuclis moderns.
- El nucli, tal com està, no obté les màximes prestacions dels nostres dispositius.
- Algunes aplicacions que volem fer servir requereixen el suport d'un nucli nou o d'algunes de les seves prestacions.
- Volem estar al dia, i ens arrisquem instal·lant últimes versions del nucli Linux.
- Ens agrada investigar o provar els nous avenços del nucli, o bé volem tocar o modificar el nucli.
- Volem programar un controlador per a un dispositiu sense suport.
- ...

Per aquests i altres motius, podem no estar contents amb el nucli que tenim. En aquest cas, se'ns plantegen dues possibilitats: actualitzar el nucli binari de la distribució o bé personalitzar-lo a partir de les fonts.

Vegem algunes qüestions relacionades amb les diferents opcions i què impliquen:

1) Actualització del nucli de la distribució. Normalment, el distribuïdor també publica les actualitzacions que sorgeixen del nucli. Quan la comunitat Linux crea una nova versió del nucli, cada distribuïdor la uneix a la seva distribució i fa les proves pertinents. Posteriorment al període de prova, s'identifiquen possibles errors, es corregeixen i es produeix l'actualització del nucli pertinent respecte a la que oferia en els CD de la distribució. Els usuaris poden baixar la nova revisió de la distribució del lloc web o bé actualitzar-la mitjançant algun sistema automàtic de paquets per mitjà d'un reposador de paquets. Normalment, es verifica quina versió té el sistema, es baixa el nucli nou i es fan els canvis necessaris perquè, la propera vegada, el sistema funcioni amb el nou nucli, i es manté la versió antiga per si hi ha problemes.

> Aquest tipus d'actualització ens simplifica molt el procés, però no ens soluciona necessàriament els problemes, ja que pot ser que el nostre maquinari no encara no tingui suport, o la característica del nucli que hem de provar encara no estigui en la versió que tenim de la distribució. Recordem que no cal fer servir l'última versió disponible (per exemple a *kernel.org*), sinó la que el distribuïdor consideri estable per a la seva distribució.

Si el nostre maquinari tampoc no ve habilitat per defecte en la nova versió, estem en la mateixa situació. O, senzillament, si volem l'última versió, aquest procés no ens serveix.

2) Personalitzar el nucli (procés descrit amb detall en els apartats següents). En aquest cas, anirem a les fonts del nucli i adaptarem "a mà" el maquinari o les característiques que vulguem. Passarem per un procés de configuració i compilació de les fonts del nucli per a, finalment, crear un nucli binari que instal·larem en el sistema i tenir-lo, així, disponible en l'arrencada següent del sistema.

També aquí ens podem trobar amb dues opcions més: o per defecte obtenim la versió "oficial" del nucli (*kernel.org*), o podem acudir a les fonts proporcionades per la pròpia distribució. Cal tenir en compte que distribucions com Debian i Fedora fan una feina important d'adequació del nucli i correcció d'errors del nucli que afecten la seva distribució. Gràcies a això, de vegades, podem disposar de correccions addicionals al codi original del nucli. Una altra vegada, les fonts que ofereix la distribució no han de correspondre necessàriament a l'última versió publicada.

Aquest sistema ens permet la màxima fiabilitat i control, però a un cost d'administració alt; ja que hem de disposar de coneixements amplis dels dispositius i de les característiques que escollim (què signifiquen i quines implicacions poden tenir), ide les conseqüències que puguin tenir les decisions que prenguem.

3. Procés de configuració i compilació

La personalització del nucli [Vasb] és un procés costós i necessita amplis coneixements del que es fa. A més, és una de les tasques crítiques, de la qual depèn l'estabilitat del sistema, per la pròpia naturalesa del nucli, ja que és l'element central del sistema.

Nota

El procés d'obtenció d'un nou nucli personalitzat passa per obtenir les fonts, adaptar la configuració, i compilar i instal·lar el nucli obtingut en el sistema.

Qualsevol error de procediment pot comportar la inestabilitat o la pèrdua del sistema. Per tant, és bo dur a terme qualsevol tasca de còpia de seguretat de les dades d'usuaris, dades de configuracions que hàgim personalitzat o, si disposem de dispositius adequats, la còpia de seguretat completa del sistema. També és recomanable disposar d'algun disquet d'arrencada (o distribució LiveCD amb eines) que ens serveixi d'ajuda si sorgeixen problemes, o bé un disquet de reparació (*rescue disk*), que la majoria de distribucions permeten crear des dels CD de la distribució (o el proporcionen directament com a CD de reparació per a la distribució).

Sense ànim d'exagerar, gairebé mai no apareixen problemes si se segueixen els passos adequadament, es té consciència del que es fa i es prenen algunes precaucions.

Vegem el procés necessari per a instal·lar i configurar un nucli Linux. Per a això, examinarem el següent:

1) El cas de les versions antigues 2.4.x.
2) Algunes consideracions sobre la migració a les 2.6.x.
3) Detalls específics de les versions 2.6.x.
4) Un cas particular per a la distribució Debian, que disposa d'un sistema propi (*debian way*) de compilació més flexible.

Les distribucions actuals, pràcticament, ja no ofereixen les versions 2.4.x, però hem de considerar que en més d'una ocasió ens veurem obligats a migrar un determinat sistema a noves versions, o bé a mantenir-lo en les antigues a causa d'incompatibilitats o presència de maquinari antic sense suport.

Els conceptes generals del procés de compilació i configuració s'explicaran en el primer apartat (2.4.x), ja que la majoria són genèrics, i després veurem les diferències respecte a les noves versions.

3.1. Compilació nucli versions 2.4.x

Les instruccions són específiques per a l'arquitectura x86 Intel, mitjançant usuari *root* (encara que part del procés es pot fer com a usuari normal):

1) Obtenir el nucli. Per exemple, podem acudir a *www.kernel.org* (o el seu servidor *ftp*) i baixar la versió que vulguem provar. Hi ha *mirrors* per a diferents

països, per exemple, podem acudir a *www.es.kernel.org*. En la majoria de les distribucions del GNU/Linux, com Fedora/Red Hat o Debian, també s'ofereix com a paquet el codi font del nucli (normalment amb algunes modificacions incloses). Si es tracta de la versió del nucli que necessitem, potser és preferible utilitzar aquestes (mitjançant els paquets *kernel-source* o similars). Si volem els últims nuclis, potser no estiguin disponibles en la distribució i haurem d'anar a *kernel.org*.

2) Desempaquetar el nucli. Les fonts del nucli se solien col·locar i desempaquetar en el directori */usr/src*, encara que es recomana utilitzar algun directori a part per a no barrejar-les amb fitxers font que pugui portar la distribució. Per exemple, si les fonts venien en un fitxer comprimit de tipus *bzip2*:

```
bzip2 -dc linux-2.4.0.tar.bz2 | tar xvf -
```

Si les fonts venien en un fitxer *gz*, reemplacem *bzip2* por *gzip*. En descomprimir les fonts, s'haurà generat un directori *linux-version_kernel*, en què entrarem per establir la configuració del nucli.

Abans de començar els passos previs a la compilació, ens hem d'assegurar de disposar de les eines correctes, en especial del compilador *gcc*, *make* i altres utilitats *gnu* complementàries en el procés. Un exemple són les *modutils,* que disposen les diferents utilitats per a l'ús i gestió dels mòduls de nucli dinàmics. D'altra banda, per a les diferents opcions de configuració, cal tenir en compte una sèrie de prerequisits en forma de llibreries associades a la interfície de configuració utilitzada (per exemple, les *ncurses* per a la interfície *menuconfig*).

En general, es recomana consultar la documentació del nucli (per mitjà de paquet o en el directori arrel de les fonts) per a conèixer quins prerequisits i versions d'aquests són necessaris per al procés. Es recomana examinar els fitxers *README* en el directori "arrel" i el Documentation/Changes, o l'índex de documentació de del nucli a Documentation/00-INDEX.

Si ja hem realitzat compilacions en el mateix directori, ens haurem d'assegurar que el directori utilitzat sigui net de compilacions anteriors. El podem netejar amb *make mrproper* (realitzat des del directori arrel).

Per al procés de configuració del nucli [Vasb], tenim diversos mètodes alternatius, que ens presenten interfícies diferents per ajustar els diversos paràmetres del nucli (que se solen emmagatzemar en un fitxer de configuració, normalment *.config*, en el directori arrel de les fonts). Les diferents alternatives són les següents:

- *make config*. Des de la línia d'ordres se'ns pregunta per cada opció, i se'ns demana confirmació (y/n) – si volem l'opció o no, o se'ns demanen els valors necessaris. Si la configuració és llarga, se'ns demanen moltes respostes

i, depenent de la versió, potser hem de respondre a gairebé un centenar de preguntes (o més, depenent de la versió).

- *make oldconfig*. Serveix per si volem reutilitzar una configuració (normalment emmagatzemada en un fitxer *.config*, en el directori arrel de les fonts). Cal tenir en compte que només és vàlida si compilem la mateixa versió del nucli, ja que diferents versions del nucli poden variar en les seves opcions.

- *make menuconfig*. És una configuració basada en menús textuals, bastant còmoda. Permet habilitar o inhabilitar el que vulguem i és més ràpida que el *make config*.

- *make xconfig*. És la més còmoda, i es basa en diàlegs gràfics en X Window. És necessari tenir instal·lades les biblioteques de *tcl/tk*, ja que aquesta configuració està programada en aquest llenguatge. La configuració es basa en quadres de diàleg i botons/caselles d'activació, es fa bastant ràpida i disposa d'ajuda amb comentaris de moltes de les opcions. Però, hi ha un defecte: pot ser que algunes de les opcions no apareguin (depèn del fet que el programa de configuració estigui actualitzat, i de vegades no ho està). En aquest últim cas, el *make config* (o el *menuconfig*) és l'únic que assegura disposar de totes les opcions elegibles. En els altres tipus de configuració depèn del fet que els programes s'hagin pogut adaptar a temps a les noves opcions quan s'allibera el nucli. Encara que, en general, s'intenten mantenir de manera equivalent.

Figura 2. Configuració del nucli (make xconfig) des d'interfície gràfica en X Window

Una vegada s'hagi fet el procés de configuració, cal desar el fitxer (*.config)*, ja que la configuració consumeix molt temps. A més, pot ser útil disposar de la configuració realitzada si està planejat fer-la en diverses màquines semblants o idèntiques.

Un altre tema important en les opcions de configuració és que, en molts casos, se'ns preguntarà si volem que una determinada característica estigui integrada

en el nucli o estigui com a mòdul (en l'apartat dedicat als mòduls en donarem més detalls). Aquesta és una decisió més o menys important, ja que el rendiment del nucli (i, per tant, del sistema sencer) de vegades depèn de la nostra elecció.

El nucli del Linux ha començat a ser gran, tant per complexitat com pels controladors (*drivers*) de dispositiu [AR01] que inclou. Si ho integréssim tot, es podria crear un fitxer del nucli força gran i ocupar molta memòria, la qual cosa alentiria alguns aspectes del funcionament. Els mòduls del nucli [Hen] són un mètode que permet separar part del nucli en petits trossos, que es carregaran a petició dinàmicament, quan siguin necessaris per càrrega explícita o per ús de la característica.

L'elecció més normal és integrar el que es consideri bàsic per al funcionament o crític en rendiment dins del nucli, i les parts o els controladors que s'hagin d'utilitzar esporàdicament o que es necessiti deixar-los com a mòduls per a futures ampliacions de l'equip.

Exemples:

- Un cas clar són els controladors de dispositiu: si estem actualitzant la màquina, pot ser que a l'hora de crear el nucli no sapiguem amb seguretat quin maquinari tindrà – per exemple, quina targeta de xarxa–, però sí sapiguem que estarà connectat a xarxa; llavors, el suport de xarxa estarà integrat en el nucli, però en els controladors de les targetes en podrem seleccionar uns quants (o tots) i posar-los com a mòduls. Així, quan tinguem la targeta, podrem carregar el mòdul necessari o, si després hem de canviar una targeta per una altra, només haurem de canviar el mòdul que es carregarà. Si només hi hagués un controlador integrat en el nucli i canviéssim la targeta, això obligaria a reconfigurar i recompilar el nucli amb el controlador de la targeta nova.

- Un altre cas (encara que no és gaire comú) és quan necessitem dos dispositius que són incompatibles entre ells, o està funcionant l'un o l'altre (això pot passar, per exemple, amb la impressora amb cable paral·lel i maquinari que es connecta al port paral·lel). Per tant, en aquest cas, hem de col·locar com a mòduls els controladors i carregar o baixar el que sigui necessari.

- Un altre exemple sn els sistemes de fitxers (*file systems*). Normalment, esperarem que el nostre sistema tingui accs a alguns d'ells, per exemple *ext2* o *ext3* (propis del Linux), o *vfat* (dels Windows 95/98/ME), i els donarem d'alta en la configuraci del nucli. Si, en un altre moment, hagussim de llegir un altre tipus no esperat, per exemple, dades desades en un disc o partici de sistema NTFS del Windows NT/XP, no podrem: el nucli no sabria o no tindria suport per fer-ho. Si tenim previst que en algun moment (per, no habitualment) s'hagi d'accedir a aquests sistemes, podem deixar els altres *file systems* com a mduls.

1) Compilació del nucli

Començarem la compilació amb el *make*. Primer cal generar les possibles dependències entre el codi, i després el tipus d'imatge de nucli que es vol (en aquest cas, una imatge comprimida, que sol ser la normal):

```
make dep
make bzImage
```

Quan aquest procés acaba, tenim la part integrada del nucli. Ens falten les parts que hàgim posat com a mòduls:

```
make modules
```

Fins a aquest moment, hem fet la configuració i compilació del nucli. Aquesta part es podia fer des d'un usuari normal o bé des del *root*, però ara necessitarem forçosament un usuari, *root* perquè passarem a la part de la instal·lació.

2) Instal·lació

Comencem instal·lant els mòduls:

```
make modules_install
```

I seguim instal·lant el nou nucli (des del directori */usr/src/linux-version* o el que hàgim utilitzat com a temporal):

```
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.0
cp System.map /boot/System.map-2.4.0
```

El fitxer *bzImage* és el nucli acabat de compilar, que es col·loca en el directori */boot*. Normalment, l'antic nucli es trobarà en el mateix directori */boot* amb el nom *vmlinuz*, o bé *vmlinuz-version-anterior* i *vmlinuz* como un enllaç simbòlic al nucli vell. Una vegada tinguem nostre nucli, és millor conservar l'antic per, si es produeixen errors o un mal funcionament en el nou, poder recuperar el vell. El fitxer *System.map* conté els símbols disponibles en el nucli i és necessari per al seu procés d'arrencada; també es col·loca en el mateix directori.

En aquest punt, també cal tenir en compte que, en l'arrencada, el nucli pot necessitar la creació de fitxers tipus *initrd*, que serveix d'imatge composta d'alguns *drivers* bàsics i s'utilitza durant la càrrega del sistema, com a primera fase, si el sistema necessita aquests *drivers* abans de procedir a l'arrencada d'alguns components. En alguns casos és imprescindible perquè, per carregar la resta del sistema, es necessiten controladors que s'hauran de carregar en una primera fase. Un exemple d'això serien controladors específics de disc, com contro-

ladors RAID o de volums, que serien necessaris per a després, en una segona fase, accedir al disc per carregar la resta del sistema.

El nucli es pot generar amb imatge *initr* o sense, segons les necessitats del maquinari o sistema concret. En alguns casos, la distribució imposa la necessitat d'utilitzar imatge *initrd;* en altres casos, dependrà del nostre maquinari. També se sol utilitzar per vigilar la mida del nucli, de manera que es pot carregar el seu bàsic per mitjà de la imatge *initrd* i, en una segona fase, carregar la resta en forma de mòduls. En cas de necessitar la imatge *initrd*, es crearia amb la utilitat *mkinitrd* (vegeu man, o taller del capítol) dins del directori */boot*.

3) El pas següent és dir al sistema amb quin nucli ha d'arrencar, encara que això depèn del sistema d'arrencada del Linux:

a) Des d'arrencada amb *lilo* [Zan][Skoa], tant si és en el MBR (*master boot record*) com des de partició pròpia, cal afegir al fitxer de configuració (en */etc/lilo.conf*), per exemple, les línies següents:

```
image = /boot/vmlinuz-2.4.0
  label = 2.4.0
```

en què *image* és el nucli que s'arrencarà, i *label* serà el nom amb què apareixerà l'opció en l'arrencada. Podem afegir aquestes línies o modificar les que hi hagués del nucli antic. Es recomana afegir-les i deixar el nucli antic per recuperar-lo si apareixen problemes. En el fitxer */etc/lilo.conf*, hi pot haver una o més configuracions d'arrencada, tant del Linux com d'altres sistemes (com el Windows).

Cada arrencada s'identifica per la seva línia *image* i el *label* que apareix en el menú d'arrencada. Hi ha una línia *default = label* en què s'indica el *label* que s'engega per defecte. També podem afegir a les línies anteriors un *root = /dev/...* per indicar la partició de disc en què hi ha el sistema de fitxers principal (el /), i recordar que els discos tenen dispositius com */dev/hda* (1r. disc *ide*) */dev/hdb* (2n. disc *ide*) o bé */dev/sdx* per a discos SCSI (o emulats), i la partició s'indicaria com a *root=/dev/hda2* si el / del nostre Linux estigués en la segona partició del primer disc *ide*. A més amb *append* =? podem afegir paràmetres a l'arrencada del nucli [Gor]. Si el sistema utilitza *initrd*, també caldrà indicar quin és el fitxer (que també estarà localitzat en */boot/initrd-versiókernel*) amb l'opció *initrd*=. Després de canviar la configuració del lilo, cal escriure-la perquè arrenqui:

```
/sbin/lilo -v
```

Reiniciem (*reboot*) i arrenquem amb el nou nucli.

Si tenim problemes, podem recuperar l'antic nucli escollint l'opció del vell nucli. Després, retocant el *lilo.conf*, podem tornar l'antiga configuració o estudiar el problema i reconfigurar i recompilar el nucli de nou.

b) En el cas de l'arrencada amb *grub* [Kan01][Pro], la gestió és bastant simple, es pot afegir una nova configuració formada pel nucli nou i sumar-la com una opció més al fitxer del *grub*. A continuació, es reinicia de manera semblant al *lilo*, però recordant que en *grub* és suficient editar el fitxer (típicament */boot/grub/menu.lst*) i reiniciar. També és millor deixar l'antiga configuració per a recuperar-se de possibles errors.

3.2. Migració a nucli 2.6.x

En cas d'haver d'actualitzar versions de distribucions antigues, o de realitzar el canvi de generació del nucli mitjançant les fonts, caldrà tenir en compte una sèrie de consideracions a causa de les novetats introduïdes en la branca 2.6.x del nucli.

A continuació fem una llista d'alguns punts concrets:

- Alguns dels mòduls del nucli han canviat de nom, i alguns han pogut desaparèixer. Cal comprovar la situació dels mòduls dinàmics que es carreguen (per exemple, examinar */etc./modulis* o */etc/modules.conf*) i editar-los per reflectir els canvis.

- S'han afegit noves opcions per a la configuració inicial del nucli com *make gconfig*, una interfície basada en *gtk* (Gnome). S'hauran de veure, en aquest cas com a prerequisit, les llibreries de Gnome. L'opció de *make xconfig* s'ha implementat ara amb les llibreries de *qt* (KDE).

- S'incrementen les versions mínimes imprescindibles de diverses utilitats necessàries per al procés de compilació (consulteu Documentation/Changese en les fonts del nucli), en especial, la versió del compilador *gcc* mínima.

- Ha canviat el paquet per defecte per a les utilitats de mòduls, que passa ser *module-init-tools* (en lloc de *modutils*, utilitzat en 2.4.x). Aquest paquet és un prerequisit per a la compilació de nuclis 2.6.x, ja que el carregador de mòduls es basa en aquesta nova versió.

- El sistema *devfs* queda obsolet en favor de l'*udev,* el sistema que controla l'arrencada (connexió) en calent (*hotplug*) de dispositius (i el seu reconeixement inicial, de fet, simulant una arrencada en calent en iniciar el sistema), i crea dinàmicament les entrades en el directori */dev*, només per als dispositius que estiguin actualment presents.

- A partir de certes versions del 2.6.x, a Debian, per a les imatges binàries de *kernels*, *headers* i codi font canvia el nom dels paquets de *kernel-images/source/headers* a *linux-image/source/headers*.

- En alguns casos, els dispositius de tecnologies noves (com SATA) poden haver passat de */dev/hdX* a */dev/sdX*. En aquests casos, caldrà editar les configuracions de */etc/fstab* i el *bootloader* (*lilo* o *grub*) per a reflectir els canvis.

- Poden sorgir alguns problemes amb dispositius d'entrada/sortida concrets. El canvi de noms de mòduls de nucli ha afectat, entre altres, els dispositius de ratolí, la qual cosa pot afectar també l'execució de l'X-Window, la verificació de quins models són necessaris i carregar els mòduls correctes (per exemple el *psmouse*). D'altra banda, en el nucli, s'integren els *drivers* de so Alsa. Si tenim els OSS antics, caldrà eliminar-los de la càrrega de mòduls, ja que Alsa ja s'encarrega de la seva emulació.

- Respecte a les arquitectures que suporta el nucli, en les diferents revisions, s'han anat incrementant les arquitectures que suporten els nuclis 2.6.x, la qual cosa ens permetrà disposar d'imatges binàries del nucli en les distribucions (o les opcions de compilació del nucli) més adequades per al suport dels nostres processadors. Concretament, podem trobar arquitectures com i386 (per a Intel i AMD), que suporten la compatibilitat d'Intel en 32 bits per a tota la família de processadors (en algunes distribucions s'utilitza 486 com a arquitectura general). En algunes distribucions s'integren versions diferenciades per a i686 (Intel a partir de Pentium Pro endavant), per a k7 (AMD Athlon d'ara endavant) i les específiques de 64 bits per a AMD 64 bits, i Intel amb extensions em64t de 64 bits com els Xeon, i Multicores. D'altra banda, també hi ha l'arquitectura IA64 per als models 64bits Intel Itanium. En la majoria dels casos, les arquitectures disposen de les capacitats SMP activades en la imatge del nucli, tret que la distribució suporti versions amb SMP o sense, creades independentment (en aquest cas, se sol afegir el sufix *-smp* a la imatge que el suporta).

- A Debian, per a la generació d'imatges *inirtrd*, a partir de certes versions del nucli (>= 2.6.12) les *mkinitrd-tools* es consideren obsoletes, i se substitueixen per noves utilitats com *initramfs-tools* o *yaird*. Totes dues permeten la construcció de la imatge *initrd*, i Debian recomana la primera.

3.3. Compilació de nucli versions 2.6.x

En les versions 2.6.x, tenint en compte les consideracions anteriors, la compilació té lloc de manera semblant a l'anterior:

Una vegada s'ha baixat el nucli 2.6.x (x és el número de revisió del nucli, o el parell de números) en el directori que es farà servir per a la compilació i s'han comprovat les versions necessàries de les utilitats bàsiques, es fa la compilació i la neteja d'anteriors compilacions:

```
# make clean mrproper
```

Configuració de paràmetres (recordem que, si disposem d'un *.config* previ, no caldrà començar de zero la configuració). La configuració la realitzem amb l'opció escollida de *make* (depenent de la interfície utilitzada):

```
# make menuconfig
```

Construcció de la imatge binària del nucli:

```
# make dep
# make bzImage
```

Construcció dels mòduls (els especificats com a tals):

```
# make modules
```

Instal·lació dels mòduls creats (*/llib/modules/version*):

```
# make modules_install
```

Còpia de la imatge en la seva posició final (suposant que i386 és l'arquitectura):

```
# cp arch/i386/boot/bzimage /boot/vmlinuz-2.6.x.img
```

I, finalment, creació de la imatge *initrd* que considerem necessària, amb les utilitats necessàries segons la versió (vegeu comentari posterior),i ajustament del *bootloader lilo* o *grub* segons quin s'utilitzi.

Els últims passos (*vmlinuz*, *system.map* i *initrd*) del moviment de fitxers a */boot* també es poden realitzar normalment amb el procés següent:

```
# make install
```

Però cal tenir en compte que realitza tot el procés, i actualitzarà els *bootloaders,* traient configuracions antigues o alterant-les; a més, es poden alterar els enllaços per defecte en el directori */boot.* Això s'ha de tenir en compte a l'hora de pensar en les configuracions passades que volem desar.

Respecte a l'*initrd*, a Fedora / Red Hat, es crearà automàticament amb l'opció *install.* A Debian haurem d'utilitzar les tècniques de l'apartat següent, o bé crear-lo expressament amb *mkinitrd* (versions < =2.6.12) o, posteriorment, amb *mkinitramfso,* una utilitat denominada *upda-teinitramfs*, i especificar la versió del nucli (s'assumeix que s'anomena *vmlinuz-version* dins del directori */boot*):

```
# update-initramfs -c -k 'version'
```

3.4. Compilació a Debian del nucli (Debian Way)

A Debian, a més dels mètodes examinats, cal afegir la configuració pel mètode denominat Debian Way, un mètode que ens permet construir el nucli d'una manera flexible i ràpida.

Per al procés, necessitarem una sèrie d'utilitats (instal·lar els paquets, o de similars): *kernel-package*, *ncurses-dev*, *fakeroot*, *wget*, *bzip2*.

Podem utilitzar el mètode amb dues finalitats: reconstruir un nucli equivalent al proporcionat per la distribució, o bé personalitzar-lo i utilitzar així el mètode per a construir un nucli equivalent personalitzat.

En el primer cas, obtenim la versió de les fonts del nucli que ens proporciona la distribució (x és la revisió del nucli 2.6):

Nota

Podeu veure el procés "Debian Way" amb detall a:
http: //kernel-handbook.alioth.debian.org/

```
# apt-get install linux-source-2.6.x
$ tar -xvjf /usr/src/linux-source-2.6.x.tar.bz2
```

En què obtenim les fonts i les descomprimim (el paquet deixa el fitxer en */usr/src*).

Instal·lació d'eines bàsiques:

```
# apt-get install build-essential fakeroot
```

Comprovació de dependències de les fonts:

```
# apt-get build-dep linux-source-2.6.x
```

I construcció del binari, segons la configuració preestablerta del paquet (semblant a la inclosa en els paquets *image* oficials del nucli a Debian):

```
$ cd linux-source-2.6.x
$ fakeroot debian/rules binary
```

Hi ha alguns procediments extres per a la creació de nuclis partint de diferents nivells de *patch* proporcionats per la distribució, i possibilitats de generar diferents configuracions finals (es pot veure la referència de la nota per a complementar aquests aspectes).

En el segon cas, més habitual, quan volem un nucli personalitzat, hem de realitzar un procés semblant amb un pas de personalització típic (per exemple, mitjançant *make menuconfig*). A més a més, hem de seguir els passos següents:

Primer hem d'obtenir i preparar el directori (aquí obtenim els paquets de la distribució, però és equivalent obtenint les fonts des de *kernel.org*):

```
# apt-get install linux-source-2.6.x
$ tar xjf /usr/src/linux-source-2.6.x.tar.bz2
$ cd linux-source-2.6.x
```

A continuació, configurem els parmetres. Com sempre, ens podem basar en fitxers *.config* que hgim utilitzat anteriorment, per a partir d'una configuraci coneguda (per a la personalitzaci, tamb es pot utilitzar qualsevol dels altres mtodes, *xconfig*, *gconfig*...):

```
$ make menuconfig
```

Construïm el nucli depenent d'*initrd* o no. Sense *initrd* disponible (cal anar amb compte amb la versió utilitzada, ja que, a partir de certa versió del nucli, pot ser obligatori l'ús d'imatge *initrd*):

```
$ make-kpkg clean
$ fakeroot make-kpkg --revision=custom.1.0 kernel_image
```

o bé si tenim d'*initrd* disponible (ja construït):

```
$ make-kpkg clean
$ fakeroot make-kpkg --initrd --revision=custom.1.0 kernel_image
```

El procés acaba amb l'obtenció del paquet associat a la imatge del nucli, que finalment podrem instal·lar:

```
# dpkg -i ../linux-image-2.6.x_custom.1.0_i386.deb
```

Afegim, també en aquest apartat, una altra peculiaritat de Debian: les utilitats per a afegir mòduls dinàmics de nucli proporcionats per tercers. En particular, la utilitat *module-assistant* permet automatitzar tot aquest procés a partir de les fonts del mòdul.

Necessitem disposar dels *headers* del nucli instal·lat (paquet *linux-headers-version*) o bé de les fonts que utilitzem en la compilació del nucli. A partir d'aquí, el *module-assistant* es pot utilitzar interactivament, i permet triar entre una àmplia llista de mòduls registrats prèviament en l'aplicació, i es pot encarregar de baixar el mòdul, compilar-lo i instal·lar-lo en el nucli.

També en la utilització des de línia d'ordres, podem especificar simplement el següent:

```
# m-a prepare
# m-a auto-install nom_modul
```

Això prepara el sistema per a possibles dependències, baixa fonts del mòdul, compila i, si no hi ha problemes, instal·la per al nucli. Podem veure el nom del mòdul en la llista interactiva de *module-assistant*.

Nota

m-a equival a *module-assistant*.

4. Apedaçar el nucli

En alguns casos, també és habitual l'aplicació de pedaços (*patch*) al nucli [lkm].

> Un fitxer de pedaç (*patch file*) respecte al nucli del Linux és un fitxer de text ASCII que conté les diferències entre el codi font original i el codi nou, amb informació addicional de noms de fitxers i línies de codi. El programa *patch* (vegeu *man patch*) serveix per a aplicar-lo a l'arbre del codi font del nucli (normalment a */usr/src*).

Els pedaços acostumen a fer falta quan un maquinari especial necessita alguna modificació en el nucli, o s'han detectat alguns errors (*bugs*) posteriors a alguna distribució àmplia d'una versió del nucli, o es vol afegir una nova prestació concreta. Per a corregir el problema (o afegir la nova prestació), se sol distribuir un pedaç en lloc d'un nucli nou sencer. Quan ja hi ha diferents pedaços d'aquesta mena, s'uneixen amb diverses millores del nucli anterior per a formar una nova versió del nucli. En tot cas, si tenim el maquinari problemàtic, o l'error afecta la funcionalitat o l'estabilitat del sistema i no podem esperar la versió següent del nucli, s'haurà d'aplicar el pedaç.

El pedaç se sol distribuir en un fitxer comprimit de tipus *bz2* (*bunzip2*, encara que també es pot trobar en *gzip* amb extensió *.gz*) com, per exemple:

```
patchxxxx-2.6.21-pversion.bz2
```

en què *xxxx* sol ser algun missatge sobre el tipus o finalitat del pedaç. 2.6.21 seria la versió del nucli al qual s'aplicarà el pedaç, i *pversion* faria referència a la versió del pedaç, del qual també n'hi poden haver diverses. Cal tenir en compte que estem parlant d'aplicar pedaços a les fonts del nucli (normalment instal·lades, com vam veure, en */usr/src/linux* o directori similar).

Una vegada disposem del pedaç, l'haurem d'aplicar. Veurem el procés a seguir en algun fitxer *readme* que acompanyi el pedaç, però, generalment, el procés segueix els passos (una vegada comprovats els requisits previs) de descomprimir el pedaç en el directori dels fitxers font i aplicar-lo sobre les fonts del nucli, com per exemple:

```
cd /usr/src/linux (o /usr/src/linux-2.6.21 o qualsevol versió).
bunzip2 patch-xxxxx-2.6.21-version.bz2
patch -p1 < patch-xxxxx-2.6.21-version
```

i, després, haurem de recompilar el nucli per a tornar-lo a generar.

Els pedaços es poden obtenir de diferents llocs. El més normal és trobar-los en el lloc de magatzem dels nuclis (*www.kernel.org*) o bé a *www.linuxhq.com*, que en té un fitxer complet. Determinades comunitats Linux (o usuaris individuals) també solen oferir algunes correccions, però és millor buscar en els llocs estàndard per assegurar un mínim de confiança en aquests pedaços, i evitar problemes de seguretat amb possibles pedaços pirates. Una altra via és el fabricant de maquinari, que pot oferir certes modificacions del nucli (o de controladors) perquè els seus dispositius funcionin millor (un exemple conegut és NVIDIA i seus *drivers* Linux per a les seves targetes gràfiques).

Finalment, assenyalarem que moltes distribucions del GNU/Linux (Fedora/Red Hat, Mandriva...) ja ofereixen nuclis apedaçats per ells mateixos i sistemes per actualitzar-los (alguns, fins i tot, de manera automàtica, com en el cas de Fedora/Red Hat i Debian). Normalment, en sistemes de producció és més recomanable seguir les actualitzacions del fabricant, encara que no ofereixi necessàriament l'últim nucli publicat, sinó el que cregui més estable per a la seva distribució, a risc de perdre prestacions d'última generació, o alguna novetat en les tècniques incloses en el nucli.

Nota

En sistemes que es vulguin tenir actualitzats, per raons de test o de necessitat de les últimes prestacions, sempre es pot acudir a www.kernel.org i obtenir el nucli més modern publicat.

5. Els mòduls del nucli

El nucli és capaç de carregar dinàmicament porcions de codi (mòduls) a petició [Hen] per a complementar la seva funcionalitat (hi ha aquesta possibilitat des de la versió 1.2 del nucli). Per exemple, els mòduls poden afegir suport per a un sistema de fitxers o per a dispositius de maquinari específics. Quan la funcionalitat que proporciona el mòdul no és necessària, aquest es pot baixar i alliberar memòria.

Normalment, a petició, el nucli identifica una característica que en aquell moment no és present en el nucli, contacta amb un *thread* del nucli anomenat *kmod* (en les versions nucli 2.0.x el dimoni s'anomenava *kerneld*), i aquest executa una ordre *modprobe* per intentar carregar el mòdul associat, a partir d'una cadena amb el nom de mòdul o bé d'un identificador genèric. Aquesta informació es consulta en el fitxer */etc/modules.conf* en forma d'àlies entre el nom i l'identificador.

A continuació es busca a

```
/lib/modules/version_kernel/modules.dep
```

per a saber si hi ha dependències amb altres mòduls. Finalment, amb el l'ordre *insmod* es carrega el mòdul des de */lib/modules/version_kernel/* (el directori estàndard per als mòduls). La *version_kernel* és la versió del nucli actual, i s'utilitza l'ordre *uname -r* per a determinar-la. Per tant, els mòduls en forma binària estan relacionats amb una versió concreta del nucli, i se solen col·locar a */lliura/modulis/version-kernel*.

Si s'han de compilar, s'ha de disposar de les fonts o dels *headers* de la versió del nucli al qual està destinat.

Nota

Els mòduls aporten una flexibilitat important al sistema i permeten que s'adapti a situacions dinàmiques.

Hi ha unes quantes utilitats que ens permeten treballar amb mòduls (solen aparèixer en un paquet de programari anomenat *modutils,* que es va reemplaçar per *module-init-tools* per a la gestió de mòduls de la branca 2.6.x):

a) *lsmod*. podem veure els mòduls carregats en el nucli (la informació s'obté del pseudofitxer */proc/modules*). Es llisten els noms, les dependències amb d'altres (entre claudàtors), la mida del mòdul en bytes, i el comptador d'ús del mòdul, que permet descarregar-lo si el compte és zero.

Exemple

Alguns mòduls en una Debian:

```
Module          Size        Used by     Tainted: P
agpgart         37.344          3       (autoclean)
apm             10.024          1       (autoclean)
```

```
parport_pc     23.304          1       (autoclean)
lp             6.816           0       (autoclean)
parport        25.992          1       [parport_pc lp]
snd            30.884          0
af_packet      13.448          1       (autoclean)
NVIDIA         1.539.872       10
es1371         27.116          1
soundcore      3.972           4       [snd es1371]
ac97_codec     10.9640         0       [es1371]
gameport       1.676           0       [es1371]
3c59x          26.960          1
```

b) *modprobe*. Intenta la càrrega d'un mòdul i de les seves dependències.

c) *insmod*. Carrega un mòdul determinat.

d) *depmod*. Analitza dependències entre mòduls i crea un fitxer de dependències.

e) *rmmod*. Treu un mòdul del nucli.

f) Altres ordres es poden utilitzar per a depuració o anàlisi dels mòduls, com *modinfo*, que llista algunes informacions associades al mòdul, o *ksyms,* (només en versions 2.4.x) que permet examinar els símbols exportats pels mòduls (també a */proc/ksyms*).

Tant si és pel mateix nucli, com per l'usuari manualment amb *insmod*, normalment, per a la càrrega s'especificarà el nom del mòdul i, opcionalment, determinats paràmetres. Per exemple, si són dispositius, és habitual especificar les adreces dels ports d'E/S o bé els recursos d'IRQ o DMA. Per exemple:

```
insmod soundx io = 0x320 irq = 5
```

6. Futur del nucli i alternatives

Els avenços en el nucli del Linux en determinats moments van ser molt ràpids, però actualment, amb una situació ja bastant estable amb els nuclis de la sèrie 2.6.x, cada vegada passa més temps entre les versions que apareixen i, en certa manera, això és bastant positiu, ja que permet tenir temps per corregir errors comesos, veure aquelles idees que no van funcionar bé i provar-ne de noves que, si resulten, s'inclouen.

En aquest apartat, comentarem algunes idees dels últims nuclis, i algunes que estan previstes, per donar indicacions sobre el futur pròxim en el desenvolupament del nucli.

Nota

El nucli continua evolucionant, incorporant les últimes novetats en suport de maquinari i millores en les prestacions.

En la sèrie anterior, sèrie 2.4.x [DBo], inclosa en la majoria de distribucions actuals, es van realitzar algunes aportacions:

- Compliment dels estàndards IEEE POSIX, que permet que molts dels programes de UNIX es puguin recompilar i executar en el Linux.

- Millor suport de dispositius com PnP, USB, Port Paral·lel, SCSI...

- Suport per a nous sistemes de fitxers, com UDF (CD-ROM reescribibles com un disc). Altres sistemes amb diari (*journal*), com els Reiser d'IBM o l'*ext3*, que permeten tenir un diari (*log* o *journal*) de les modificacions dels sistemes de fitxers, i així es poden recuperar d'errors o tractaments incorrectes dels fitxers.

- Suport de memòria fins a 4 GB, al seu dia van sorgir alguns problemes amb nuclis 1.2.x que no suportaven més de 128 MB de memòria (en aquell temps era molta memòria).

- Es va millorar la interfície */proc*. Aquest és un pseudosistema de fitxers (el directori */proc*) que no existeix realment en disc, sinó que simplement és una manera d'accedir a dades del nucli i del maquinari d'una manera organitzada.

- Suport del so en el nucli. Es van afegir parcialment els controladors Alsa que abans es configuraven separadament.

- Es va incloure suport preliminar per al RAID de programari, i el gestor de volums dinàmics LVM1.

En sèrie actual, la branca del nucli 2.6.x [Pra], va disposar d'importants avenços respecte a l'anterior (amb les diferents revisions *.x* de la branca 2.6):

- Millors prestacions en SMP, important per a sistemes multiprocessadors molt utilitzats en entorns empresarials i científics.

- Millores en el planificador de CPU (*scheduler*).

- Millores en el suport *multithread* per a les aplicacions d'usuari. S'incorporen nous models de *threads* NGPT (IBM) i NPTL (Red Hat) (amb el temps, finalment es va consolidar la NPTL).

- Suport per a USB 2.0.

- Controladors Alsa de so incorporats en el nucli.

- Noves arquitectures de CPU de 64 bits, que suporten AMD x86_64 (també coneguda com a *amd64*) i PowerPC 64, i IA64 (arquitectura dels Intel Itanium).

- Sistemes de fitxers amb diari: JFS, JFS2 (IBM), i XFS (Silicon Graphics).

- Millores de prestacions en E/S, i nous models de controladors unificats.

- Millores en implementació TCP/IP, i sistema NFSv4 (compartició del sistema de fitxers per xarxa amb altres sistemes).

- Millores significatives per a nucli apropiatiu: permet que, internament, el nucli gestioni diverses tasques que es poden interrompre entre elles, imprescindible per a implementar eficaçment sistemes de temps real.

- Suspensió del sistema i restauració després de reiniciar (per nucli).

- UML, User Mode Linux, una mena de màquina virtual del Linux sobre el Linux que permet veure un Linux (en mode usuari) executant-se sobre una màquina virtual. Això és ideal per a la pròpia depuració, ja que es pot desenvolupar i verificar una versió del Linux sobre un altre sistema. Això és útil tant per al propi desenvolupament del nucli, com per a fer-ne una anàlisi de seguretat.

- Tècniques de virtualització incloses en el nucli: en les distribucions s'han anat incorporant diferents tècniques de virtualització, que necessiten extensions en el nucli. Destaquem, per exemple, nuclis modificats per a Xen o Virtual Server (Vserver).

- Nova versió del suport de volums LVM2.

- Nou pseudosistema de fitxers /*sys*, destinat a incloure la informació del sistema, i dispositius que aniran migrant des del sistema /*proc*, i deixaran aquest últim amb informació relacionada amb els processos, i el seu desenvolupament en execució.

- Mòdul FUSE per a implementar sistemes de fitxers en espai d'usuari.

En el futur, es té pensat millorar els aspectes següents:

- Increment de la tecnologia de virtualització en el nucli, per a suportar diferents configuracions d'operatius i diferents tecnologies de virtualització, i per millorar el suport del maquinari per a virtualització inclòs en els processadors que sorgeixin en les noves arquitectures.
- El suport d'SMP (màquines multiprocessador), de CPU de 64 bits (Itanium d'Intel, i Opteron d'AMD), el suport de CPU multiCore.
- La millora de sistemes de fitxers per a clusterització (*clustering*) i sistemes distribuïts.
- Al contrari, la millora per a nuclis més optimitzats per a dispositius mòbils (PDA, telèfons...).
- Millora en el compliment dels estàndards POSIX, etc.
- Millora de la planificació de la CPU. Encara que en la sèrie inicial de la branca 2.6.x es van fer molts avenços en aquest aspecte, en algunes situacions encara s'obté un rendiment baix, en particular en l'ús d'aplicacions interactives d'escriptori. S'estan estudiant diferents alternatives, per a millorar aquest i altres aspectes.

Nota

POSIX:
www.UNIX-systems.org/

A més a més, encara que s'aparta dels sistemes Linux, l'FSF (Free Software Foundation) i el seu projecte GNU continuen treballant per acabar un sistema operatiu complet. Recordem que el projecte GNU tenia com a principal objectiu aconseguir un clon Unix de programari lliure, i les utilitats GNU només són el programari de sistema necessari. A partir de 1991, quan Linus aconsegueix conjuntar el seu nucli amb algunes utilitats GNU, es fa un primer pas que ha acabat en els sistemes GNU/Linux actuals. Però el projecte GNU continua treballant en la seva idea d'acabar el sistema complet. En aquest moment, ja disposen d'un nucli en el qual poden córrer les seves utilitats GNU. A aquest nucli s'anomena Hurd, i un sistema construït amb ell es coneix com a GNU/Hurd. Ja hi ha algunes distribucions de prova, en concret, una Debian GNU/Hurd.

Nota

El projecte GNU:
www.gnu.org/gnu/thegnuproject.html

El Hurd va ser pensat com el nucli per al sistema GNU cap al 1990, quan va començar el seu desenvolupament, ja que llavors gairebé tot programari GNU estava desenvolupat, i només faltava el nucli. Va ser el 1991 quan Linus va combinar el GNU amb el seu nucli Linux, i així va crear l'inici dels sistemes GNU/Linux. Però el Hurd es continua desenvolupant. Les idees de desenvolupament en el Hurd són més complexes, ja que el Linux es podria considerar un disseny "conservador", que partia d'idees ja conegudes i implantades.

Nota

GNU i Linux, de Richard Stallman:
www.gnu.org/gnu/linux-and-gnu.html

En concret, el Hurd estava pensat com una col·lecció de servidors implementats sobre un *microkernel* Mach [Vah96], el qual és un disseny de nucli de tipus

microkernel (a diferència del Linux, que és de tipus monolític) desenvolupat per la Universitat de Carnegie Mellon i, posteriorment, per la d'Utah. La idea base era modelar les funcionalitats del nucli del UNIX com a servidors que s'implementarien sobre un nucli bàsic Mach. El desenvolupament del Hurd es va retardar mentre s'acabava el disseny del Mach, i aquest es va publicar finalment com a programari lliure, que permetria utilitzar-lo per desenvolupar el Hurd. Recordem, en aquest punt, la importància del Mach, ja que molts operatius s'han basat en idees seves; el més destacat és el MacOS X d'Apple.

El desenvolupament del Hurd es va retardar més per la complexitat interna, ja que hi havia diversos servidors amb diferents tasques de tipus *multithread* (d'execució de múltiples fils), i la depuració era extremadament difícil. Però avui en dia, ja es disposa de les primeres versions de producció, i de versions de prova de distribució GNU/Hurd.

Pot ser que, en un futur no tan llunyà, coexisteixin sistemes GNU/Linux amb GNU/Hurd, o fins i tot sigui substituït el nucli Linux per l'Hurd, si prosperen algunes demandes judicials contra el Linux (llegiu cas SCO davant IBM), ja que seria una solució per a evitar problemes posteriors. En tot cas, tant uns sistemes com els altres tenen un futur prometedor. El temps dirà cap a on s'inclina la balança.

7. Taller: configuració del nucli per a les necessitats de l'usuari

En aquest apartat, veurem un petit taller interactiu per al procés d'actualització i configuració del nucli en el parell de distribucions utilitzades: Debian i Fedora.

Una primera cosa imprescindible, abans de començar, és conèixer la versió actual que tenim del nucli amb *uname -r*, per poder determinar quina és la versió següent que volem actualitzar o personalitzar. I la segona cosa imprescindible és de disposar de mitjans per a engegar el nostre sistema en cas d'errors: el conjunt de CD de la instal·lació, el disquet (o CD) de reparació (actualment se sol utilitzar el primer CD de la distribució), o alguna distribució en LiveCD que ens permeti accedir al sistema de fitxers de la màquina per refer configuracions que hagin causat problemes. A més, cal fer còpia de seguretat de les nostres dades o configuracions importants.

Veurem les possibilitats següents:

1) Actualització del nucli de la distribució. Cas automàtic de Debian.
2) Actualització automàtica a Fedora.
3) Personalització d'un nucli genèric (tant Debian com Fedora). En aquest últim cas, els passos són bàsicament els mateixos que els que es presenten en l'apartat de configuració, però farem alguns comentaris més.

7.1. Actualitzar nucli a Debian

En el cas de la distribució Debian, la instal·lació també es pot fer automàticament, mitjançant el sistema de paquets d'APT. Es pot fer tant des de línia d'ordres com amb gestors APT gràfics (*synaptic*, *gnome-apt*...).

Efectuarem la instal·lació per línia d'ordres amb *apt-get*, suposant que l'accés a les fonts *apt* (sobretot als Debian originals) està ben configurat en el fitxer de */etc/apt/sources.list*. Vegem-ne els passos:

1) Actualitzar la llista de paquets:

```
# apt-get update
```

2) Llistar paquets associats a imatges del nucli:

```
# apt-cache search linux-image
```

3) Escollir una versió adequada a la nostra arquitectura (genèrica, 386/486/686 per a Intel, k6 o k7 per a *amd*, o en particular per a 64Bits versions amd64, Intel

i *amd*, o ia64, per a Intel Itanium). La versió va acompanyada de versió nucli, revisió de Debian del nucli i arquitectura. Per exemple: 2.6.21-4-k7, nucli per a AMD Athlon, revisió Debian 4 del nucli 2.6.21.

4) Comprovar que hi hagi els mòduls accessoris extres (amb el mateix número de versió) per a la versió escollida. Amb *apt-cache* busquem si hi ha altres mòduls dinàmics interessants per al nostre maquinari, segons la versió del nucli que volem instal·lar. Recordem que, com vam veure en la Debian Way, també hi ha la utilitat *module-assistant*, que ens permet automatitzar aquest procés després de la compilació del nucli. Si els mòduls necessaris no fossin suportats, això ens podria impedir actualitzar el nucli si considerem que el funcionament del maquinari problemàtic és vital per al sistema.

5) Buscar, si també volem disposar del codi font del nucli, els *Linux-source-version* (només el 2.6.21, és a dir, els números principals) i els *kernel-headers* corresponents, per si més tard volem fer un nucli personalitzat: en aquest cas, el nucli genèric corresponent apedaçat per Debian.

6) Instal·lar el que hàgim decidit, si volem compilar des de les fonts o simplement disposar del codi:

```
# apt-get install linux-image-version
# apt-get install xxxx-modules-version (si fossin necessaris
alguns mòduls)
```

i

```
# apt-get install linux-source-version-generica
# apt-get install linux-headers-version
```

7) Instal·lar el nou nucli, per exemple, en el carregador de l'arrencada (*bootloader*) *lilo*. Normalment, això es fa automàticament. Si se'ns pregunta si tenim l'*initrd* activat, caldrà verificar el fitxer de *lilo* (*/etc/lilo.conf*) i incloure, en la configuració *lilo* de la imatge nova, la nova línia:

initrd = /initrd.img-version (o /boot/initrd.img-version)

una vegada feta aquesta configuració, hauríem de tenir un *lilo* del mode (fragment), suposant que *initrd.img* i *vmlinuz* siguin enllaços a la posició dels fitxers del nou *kernel*:

```
default = Linux
image = /vmlinuz
      label = Linux
      initrd = /initrd.img
#     restricted
#     alias = 1
image = /vmlinuz.old
```

```
        label = LinuxOLD
        initrd = /initrd.img.old
#       restricted
#       alias = 2
```

Tenim la primera imatge per defecte, l'altra és el nucli antic. Així, des del menú *lilo* podrem demanar l'una o l'altra, o recuperar l'antiga simplement canviant el *default*. Sempre que realitzem canvis en */etc/lilo.conf* no hem d'oblidar reescriure'ls en el sector corresponent amb l'ordre */sbin/lilo* o */sbin/lilo -v*.

7.2. Actualitzar el nucli a Fedora / Red Hat

L'actualització del nucli en la distribució Fedora/Red Hat és totalment automàtica, i es realitza per mitjà del seu servei de gestió de paquets o bé pels programes gràfics que inclou la distribució per a l'actualització; per exemple, en els Red Hat empresarials n'hi ha un anomenat *up2date*. Normalment, el trobarem en la barra de tasques o en el menú d'eines de sistema de Fedora/Red Hat.

Aquest programa d'actualització, bàsicament, verifica els paquets de la distribució actual davant una base de dades de Fedora/Red Hat, i ofereix la possibilitat de baixar els paquets actualitzats, entre ells els del nucli. Aquest servei de Red Hat empresarial funciona per un compte de servei, i Red Hat l'ofereix de pagament. Amb aquest tipus d'utilitats, l'actualització del nucli és automàtica.

Per exemple, en la figura 10, veiem que una vegada iniciada l'execució ens ha detectat una nova versió del nucli disponible i la podem seleccionar perquè ens la baixi:

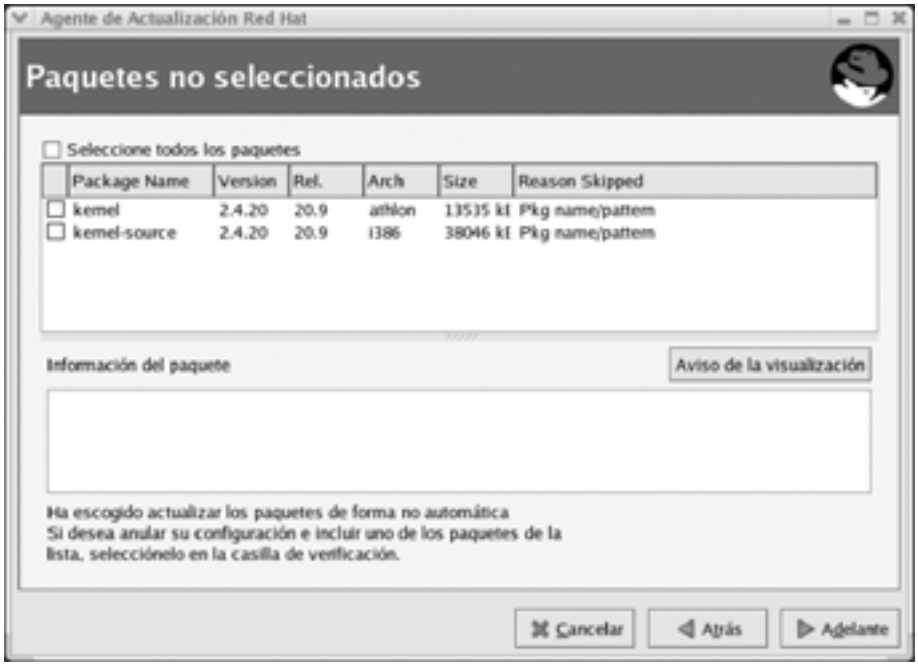

Figura 3. El servei d'actualització de Red Hat (Red Hat Network up2date) mostra l'actualització del nucli disponible i les seves fonts

A Fedora, podem utilitzar les eines gràfiques equivalents, o el *yum* directament, si coneixem la disponibilitat de nous nuclis:

```
# yum install kernel kernel-source
```

Un cop baixada, la instal·larem, normalment, també de manera automàtica, tant si disposem de *grub* com *lilo*, com a gestors d'arrencada. En el cas de *grub*, sol ser automàtic i deixa un parell d'entrades en el menú, una per a la versió més nova i una altra per a l'antiga.

Per exemple, en aquesta configuració de *grub* (el fitxer és a */boot/grub/grub.conf* o bé */boot/grub/menu.lst*), tenim dos nuclis diferents, amb els números de versió respectius:

```
#fitxer grub.conf
default = 1
timeout = 10
splashimage = (hd0,1)/boot/grub/splash.xpm.gz

title Linux (2.6.20-2945)
root (hd0,1)
kernel /boot/vmlinuz-2.6.20-2945 ro root = LABEL = /
initrd /boot/initrd-2.6.20-18.9.img

title LinuxOLD (2.6.20-2933)
root (hd0,1)
kernel /boot/vmlinuz-2.4.20-2933 ro root = LABEL = /
initrd /boot/initrd-2.4.20-2933.img
```

Cada configuració inclou un títol, que apareixerà en l'arrencada; l'arrel, o partició del disc des d'on arrencar; el directori en què es troba el fitxer corresponent al *kernel*, i el fitxer *initrd* corresponent.

Si disposem de *lilo* (per defecte s'utilitza *grub*) en la Fedora/Red Hat com a gestor, el sistema també l'actualitza (fitxer */etc/lilo.conf*), però després caldrà reescriure l'arrencada amb l'ordre */sbin/lilo* manualment.

També es pot assenyalar que, amb la instal·lació anterior, teníem possibilitats de baixar-nos les fonts del nucli. Aquestes, una vegada instal·lades, són a */usr/src/linux-version*, i es poden configurar i compilar pel procediment habitual com si fos un nucli genèric. Cal esmentar que l'empresa Red Hat du a terme un gran treball de pedaços i correccions per al nucli (utilitzat després a Fedora), i que seus nuclis són modificacions de l'estàndard genèric amb bastants afegits, per la qual cosa pot ser millor utilitzar les fonts pròpies de Red Hat, tret que vulguem un nucli més nou o experimental que el que ens proporcionen.

7.3. Personalitzar i instal·lar un nucli genèric

Veurem el cas general d'instal·lació d'un nucli a partir de les seves fonts. Suposem que tenim unes fonts ja instal·lades en /usr/src (o el prefix corresponent). Normalment, tindrem un directori *linux*, o *linux-version* o, senzillament, el número version; aquest serà l'arbre de les fonts del nucli.

Aquestes fonts poden provenir de la mateixa distribució (o que les hàgim baixat en una actualització prèvia). Primer serà interessant comprovar si són les últimes disponibles, com ja hem fet abans amb Fedora o Debian. O si volem tenir les versions últimes i genèriques, podem anar a *kernel.org* i baixar l'última versió disponible, i millor l'estable que les experimentals, tret que estiguem interessats en el desenvolupament del nucli. Baixem el fitxer i descomprimim les fonts del nucli a */usr/src* (o, gairebé millor, un altre escollit). També podríem buscar si hi ha pedaços per al nucli i aplicar-los (segons hem vist en l'apartat 4).

A continuació, comentarem els passos que caldrà realitzar. Ho farem breument, ja que molts els hem estudiat anteriorment quan treballàvem la configuració i personalització.

Nota

Seria convenient rellegir l'apartat 3.3.

1) Netejar el directori de proves anteriors (si escau):

```
make clean mrproper
```

2) Configurar el nucli, per exemple, amb *make menuconfig* (o *xconfig*, *gconfig* o *oldconfig*). Ho vam veure en l'apartat 3.

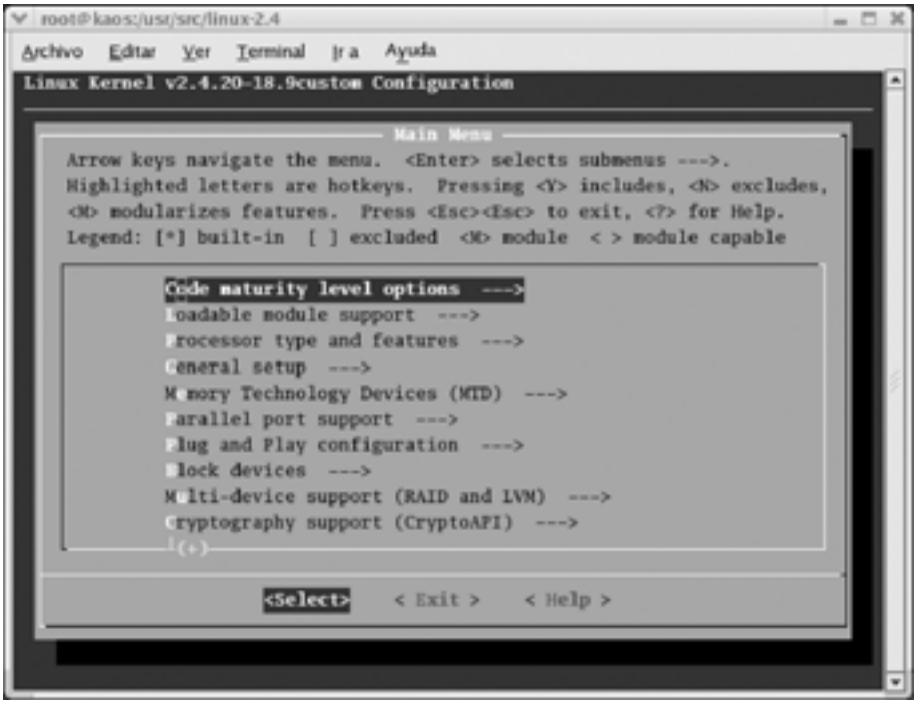

Figura 4. Configuració del nucli per menús textuals

3) Dependències i neteja de compilacions anteriors:

```
make dep
```

4) Compilació i creació de la imatge del nucli: *make bzImage*. També seria possible *zImage* si la imatge fos més petita, però és més normal *bzImage*, que optimitza el procés de càrrega i la compressió per a nuclis més grans. És possible que, en algun maquinari molt antic, no funcioni i sigui necessari *zImage*. El procés pot durar des d'uns quants minuts a una hora en maquinari modern (CPU d'1-3 GHz) i hores en maquinari molt antic. Quan acaba, la imatge es troba en */usr/src/directori-fuentes/arch/i386/boot*.

5) Ara compilem els mòduls amb *make modules*. Fins a aquest moment no hem modificat res en el nostre sistema. Ara haurem de fer la instal·lació.

6) En el cas dels mòduls, si provem alguna versió antiga del nucli (branca 2.2 o primeres de la 2.4) cal anar amb compte, ja que en alguna es sobreescrivien els antics (en les últimes 2.4.x o 2.6.x ja no és així).

Però, compte també, si compilem la mateixa versió (exacta numeració) que la que tenim (els mòduls se sobreescriuran). En aquest cas, és millor fer una còpia de seguretat dels mòduls:

```
cd /lib/modules
tar -cvzf old_modules.tgz versionkernel-antiga/
```

Així, tenim una versió en *.tgz* que podem recuperar en cas de problemes. I, finalment, instal·lem els mòduls:

```
make modules install
```

7) Ara podem passar a la instal·lació del nucli, per exemple, amb el següent:

```
# cd /usr/src/directori-Fonts/arch/i386/boot
# cp bzImage /boot/vmlinuz-versionkernel
# cp System.map /boot/System.map-versionkernel
# ln -s /boot/vmlinuz-versionkernel /boot/vmlinuz
# ln -s /boot/System.map-versionkernel /boot/System.map
```

Així col·loquem el fitxer de símbols del nucli (*System.map*) i la imatge del nucli.

8) Ja només ens queda posar la configuració necessària en el fitxer de configuració del gestor d'arrencada, tant si és *lilo* (*/etc/lilo.conf*) com *grub* (*/boot/grub/grub.conf*), segons les configuracions que ja vam veure amb Fedora o Debian. I recordem que, en el cas de *lilo*, caldrà tornar a actualitzar la configuració amb */sbin/lilo* o */sbin/lilo -v*.

9) Reiniciar la màquina i observar els resultats (si tot ha anat bé).

Activitats

1. Determineu la versió actual del nucli Linux incorporada en la nostra distribució. Comproveu les actualitzacions disponibles de manera automàtica, a Debian (*apt*) o a Fedora/Red Hat (via *yum* o *up2date*).

2. Actualitzeu automàticament la nostra distribució. Comproveu possibles dependències amb altres mòduls utilitzats (*pcmcia* o d'altres) i amb el carregador de l'arrencada (*lilo* o *grub*) utilitzat. Es recomana fer una còpia de seguretat de les dades importants del sistema (comptes d'usuaris i fitxers de configuració modificats), o bé fer-la en un altre sistema que fem servir per a proves.

3. Per a la nostra branca del nucli, determineu l'última versió disponible (consulteu *www.kernel.org*), i realitzeu una instal·lació manual amb els passos examinats en la unitat. La instal·lació final és opcional, o es pot posar una entrada dins del carregador de l'arrencada per a les proves del nou nucli.

4. En el cas de la distribució Debian, a més dels passos manuals, ja vam veure que hi havia una manera especial (recomanada) d'instal·lar el nucli a partir de les seves fonts mitjançant el paquet *kernel-package*.

Altres fonts de referncia i informació

[Kerb] Lloc que proporciona un magatzem de les diverses versions del nucli Linux i els seus pedaços.

[Kera] [lkm] Llocs web que recullen una part de la comunitat del nucli de Linux. Disposa de diversos recursos de documentació i llistes de correu de l'evolució del nucli, la seva estabilitat i les noves prestacions que es desenvolupen.

[DBo] Llibre sobre el nucli de Linux 2.4, que detalla els diferents components i la seva implementació i disseny. Hi ha una primera edició sobre el nucli 2.2 i una nova actualització al nucli 2.6.

[Pra] Article que descriu algunes de les principals novetats de la nova sèrie 2.6 del nucli Linux.

[Ker] [Mur] Projectes de documentació del nucli incomplets però amb material útil.

[Bac86] [Vah96] [Tan87] Alguns textos sobre els conceptes, disseny i implementació dels nuclis de diferents versions Unix.

[Skoa][Zan01][Kan][Pro] Per a tenir més informació sobre els carregadors *lilo* i *grub*.