Approfondimenti su Linux

La storia

Il sistema operativo Linux e' stato inizialmente creato da uno studente, Linus Torvalds, all' Universita' di Elsinky in Finlandia.

Era il 1991 e debutto' la release 0.02, la prima release stabile arrivo' nel 1994 con la 1.0.

Oggi siamo arrivati alla release stabile 2.6.9
(Linux Kernel Archives: http://www.kernel.org)

Il codice del kernel e' sviluppato e distribuito con GNU General Public License; il codice sorgente e' quindi pubblico

Inoltre sono stati sviluppati degli *standard* per documentare ed uniformare struttura e codice di Linux: http://www.linuxbase.org

Il Kernel Linux

Il kernel di linux consiste di diverse parti importanti:
la gestione dei processi, la gestione della memoria,
i driver per i dispositivi hardware, i driver per i filesystem,
la gestione della rete ed altre parti minori

a gestione dei processi crea i processi ed implementa il multitasking cambiando il processo attivo sul processore.
La gestione della memoria si occupa di assegnare le aree di memoria e di spazio di swap ai processi e a parti del kernel
Al livello più basso, il kernel contiene un driver per ciascun dispositivo hardware che supporta.

Il Kernel

Alcuni dei servizi software forniti dal kernel stesso hanno proprietà simili, e possono dunque essere astratti in classi. Ad esempio, i vari protocolli di rete sono stati astratti in un'interfaccia di programmazione, la BSD socket library.

Un altro esempio è il livello filesystem virtuale(VFS), che astrae le operazioni su filesystem, senza considerare la loro implementazione.

Ciascun tipo di filesystem fornisce un'implementazione di ogni operazione; quando un'entità prova ad usare un filesystem, la richiesta passa per il VFS, che la gira al driver opportuno .

Dipartimento di Informatica e Scienze dell'Informazione

Le fasi del boot di Linux

1- All'accensione il BIOS su ROM non volatile definisce l'ordine dei device da utilizzare per effettuare il boot.

2- Il BOOT SECTOR del primo device di boot contiene il codice (o i riferimenti su dove trovarlo) del loader che esegue il bootstrap del sistema operativo. Nel caso di Linux: LILO o GRUB.

3- Il loader lancia il caricamento del kernel di Linux, che si copia in memoria ed esegue i controlli e il riconoscimento dell'hardware presente.

4- A fine caricamento il kernel esegue il processo init, padre di tutti i processi, che gestisce il caricamento di tutti gli altri programmi da eseguire per completare il boot.

Dipartimento di Informatica e Scienze dell'Informazione

BIOS

Ogni sistema Intel ha sulla motherboard un BIOS su ROM con cui gestire l'hardware del sistema.

All'avvio di un computer non c'è nulla in RAM e nessun programma predefinito da caricare.

Le istruzioni su come procedere sono nella memoria non volatile del BIOS, in cui, fra le impostazioni definibili dall'utente, c'e' la sequenza dei dispositivi da usare per il boot.

Nei BIOS più recenti è possibile fare il boot da floppy, cdrom, hard disk (potendo scegliere se dare precedenza a HD IDE o SCSI) e altri dispositivi quali Zip o scheda di rete.

Dipartimento di Informatica e Scienze dell'Informazione

Boot sector e Master Boot Record

Il BIOS cerca, nell'ordine configurato, il boot sector sui diversi dispositivi di boot previsti.

Gli hard disk hanno un boot sector per ogni partizione e un unico Master Boot Record (MBR) che è il primo boot sector dell'intero hard disk.

Se si esegue il boot da un hard disk, è il codice contenuto nel MBR che viene eseguito.

Dipartimento di Informatica e Scienze dell'Informazione

Loader

Esistono diversi loader che eseguono il bootstrap del sistema operativo per Linux:

Su sistemi Intel Based: LILO, GRUB, LOADLIN, SYSLINUX, BOOTLIN;
su sistemi Alpha: MILO;
su sistemi Sparc: SILO.

Tutti di fatto eseguono la stessa funzione, alcuni (loadlin e syslinux) sono programmi DOS che eseguono il kernel caricandolo da una partizione DOS, altri sono adattamenti di LILO che riguardano sistemi non basati su processori Intel.

Inizializzazione del Sistema

Il kernel, invocato dal loader, viene caricato in memoria

Il kernel di Linux è installato compresso, quindi per prima cosa si decomprime. L'inizio dell'immagine del kernel contiene un programmino che fa proprio questo

Successivamemente inizializza i vari device driver

L'ultima operazione eseguita dal kernel alla fine del suo caricamento è il lancio del processo init, il padre di tutti i processi.

Da questo momento tutto il codice eseguito lavora in user space (in kernel space lavorano solo il kernel e i suoi moduli).

I principali servizi: INIT

INIT

Il servizio più importante di un sistema UNIX è fornito da init.

init viene inizializzato come primo processo di ciascun sistema UNIX, ed è l'ultima cosa che il kernel fa all'avvio.

Init continua il processo di boot (controlla e monta i filesystem, avvia i demoni ecc.), si assicura che stiano girando i processi di getty multipli (per poter far collegare gli utenti), e adotta i processi orfani

Quando il sistema viene spento, è init che deve uccidere tutti gli altri processi, smontare i filesystem e fermare il processore

I principali servizi

Login

Il login dai terminali (attraverso linee seriali) e dalla console (quando non si sta usando X) vengono forniti dal programma getty. init avvia una copia di getty per ogni terminale da cui sono consentiti i login.

Il programma getty legge il nome dell'utente e avvia il programma login, che legge la password. Se il nome dell'utente e la password sono corretti, login avvia la shell.

Quando la shell termina, cioè l'utente si scollega, o quando login termina perché il nome dell'utente e la password non corrispondono, init lo nota e avvia un'altra copia di getty.

Il kernel non ha nozione dei login, che vengono tutti gestiti dai programmi di sistema.

I principali servizi

Syslog

Il kernel e molti programmi di sistema producono messaggi di errore, di avvertimento e di altro tipo. Spesso è importante che questi messaggi possano essere letti in un secondo tempo, anche dopo parecchio, quindi devono essere scritti in un file.

Il programma che lo fa è syslog, che può essere configurato per distribuire i messaggi in file diversi a seconda di chi li genera o del loro grado di importanza.

Ad esempio, i messaggi del kernel sono spesso rediretti in un file separato dagli altri, dato che i messaggi del kernel spesso sono più importanti e devono essere letti regolarmente per individuare i problemi.

I principali servizi

Cron e at

Sia gli utenti che gli amministratori di sistema hanno spesso bisogno di avviare periodicamente dei comandi.

Ad esempio, l'amministratore di sistema potrebbe voler avviare un programma che ripulisca le directory che contengono file temporanei

Il servizio di cron funziona proprio per questo. Ciascun utente ha una *crontab*, dove elenca i comandi che vuole eseguire ed il momento in cui farlo; il demone cron avvia poi i comandi nel momento specificato.

Il servizio *at* è simile a cron, ma vale per una sola volta: il comando viene eseguito al momento indicato, ma non viene ripetuto.

I principali servizi

Graphical User Interface – GUI

UNIX e Linux non hanno incorporata l'interfaccia utente nel kernel; invece, la fanno implementare dai programmi a livello utente.

Questo si applica sia per l'ambiente testuale che per quello grafico.Questo modo di agire rende il sistema più flessibile, ma ha lo svantaggio di facilitare l'implementazione di un'interfaccia diversa per ciascun programma, rendendo il sistema più difficile da imparare.

L'ambiente grafico usato principalmente con Linux si chiama Sistema X Window (in breve, X). X stesso non implementa un'interfaccia utente, ma solo un sistema di primitive grafiche per il tracciamento di finestre, cioè degli strumenti con cui implementarle.

I tre stili più conosciuti in X sono Athena, Motif e Open Look.

Gnome usa GTK e kde usa QT

I principali servizi

Login in rete

Ci sono diversi modi possibili per collegarsi attraverso una rete: nelle reti di tipo TCP/IP troviamo: telnet, rlogin e ssh.

Invece di un insieme di getty i login di rete usano un demone per ciascuna modalità di collegamento che stanno in ascolto per i tentativi di login in ingresso.

Quando ricevono una richiesta di connessione, inizializzano una copia di se stessi per gestire quel singolo tentativo; l'istanza originale continua ad aspettare altre richieste.

La nuova istanza funziona in modo simile a getty.

I principali servizi

File system condivisi

Una delle cose più utili che possono essere fatte con i servizi di rete è la condivisione di file attraverso un filesystem di rete.

Quello che viene usato di solito si chiama Network File System, o NFS, ed è sviluppato dalla SUN.

Con un filesystem di rete qualsiasi operazione su file fatta da un programma su una macchina è spedita attraverso la rete ad un altro computer.

Questo inganna il programma che utilizza i file considerandoli locali, mentre in realtà risiedono su un computer remoto: un modo semplice per la condivisione di informazioni, che non richiede modifiche ai programmi.

I principali servizi

La posta elettronica

Ciascun utente ha una casella di posta in arrivo (un file nel formato speciale) dove viene immagazzinata tutta la nuova posta in arrivo. Quando qualcuno spedisce della posta, il programma di posta ritrova la casella del destinatario e aggiunge il messaggio al file.

Il sistema di posta e' realizzato mediante svariati programmi. La distribuzione a caselle locali o remote viene fatta da un programma mail transfer agent o MTA (ad es. sendmail o smail), mentre i programmi utilizzati dagli utenti sono molti e vari (ad es. pine, elm, netscape).

Le caselle di posta vengono in genere tenute in /var/spool/mail

La struttura del filesystem

Esiste uno standard dei filesystem Linux, FSSTD versione 1.2 che tenta di impostare uno standard per l'organizzazione dell'albero delle directory nei sistemi Linux.

Uno standard del genere ha il vantaggio di rendere piu' agevole la scrittura o il porting del software per Linux, e amministrare le macchine Linux, poiche' i file si troveranno nel posto designato.

Non c'è nessuna autorità che impone a nessuno di uniformarsi allo standard, ma questo ha il supporto della maggior parte, se non di tutte, le distribuzioni Linux.

Non è una buona idea rompere con lo standard FSSTND se non per ragioni molto particolari. Il FSSTND tenta di seguire la tradizione Unix e le tendenze più recenti, rendendo i sistemi Linux familiari per chi ha esperienza con altri sistemi Unix, e viceversa.

La struttura del filesystem

Il filesystem *root* è specifico per ciascuna macchina (generalmente viene immagazzinato su un disco locale, anche se può trattarsi di un ramdisk o di un disco in rete) e contiene i file necessari per avviare il sistema e per portarlo ad uno stato tale che possa montare gli altri filesystem.

Il contenuto del filesystem di root sarà quindi sufficiente per la modalità a singolo utente. Conterrà anche gli strumenti per recuperare un filesystem danneggiato o copiare dai backup i file perduti. Il filesystem root dovrebbe generalmente essere piccolo, dato che contiene file estremamente critici, e un filesystem piccolo che viene modificato poco ha migliori possibilità di non venire corrotto.

Un filesystem di root corrotto in genere significa che diventa impossibile avviare il sistema tranne che con misure eccezionali (ad esempio da un floppy), quindi è meglio non rischiare.

La struttura del filesystem

La directory principale in genere non contiene nessun file, tranne forse l'immagine standard di avvio per il sistema, che di solito si chiama /vmlinuz. Tutti gli altri file sono in sottodirectory del filesystem root

Dipartimento di Informatica e Scienze dell'Informazione

La struttura del filesystem

/usr

Il filesystem /usr contiene tutti i comandi, le librerie, le pagine man e altri file che non vengono modificati durante le normali operazioni.

Nessun file in /usr dovrebbe essere specifico per nessuna macchina data, né dovrebbe essere modificato durante il normale uso. Questo permette che il file venga condiviso in rete, cosa che può portare ad un risparmio economico dato che permette di risparmiare in spazio disco e può rendere l'amministrazione molto più semplice (basta modificare solo /usr principale quando si aggiorna un'applicazione, e non c'e' bisogno di farlo separatamente su ciascuna macchina).

Anche se il filesystem si trova su un disco locale, può essere montato con accesso a sola lettura, per diminuire le possibilità di corruzione durante un crash.

Dipartimento di Informatica e Scienze dell'Informazione

La struttura del filesystem

/usr/local

Il filesystem /usr è spesso grande, dato che vi sono installati tutti i programmi.

Tutti i file in /usr vengono di solito da una distribuzione di Linux; i programmi installati in locale e il resto vanno sotto /usr/local; in questo modo è possibile aggiornare il sistema a una nuova versione della distribuzione, o addirittura ad una distribuzione completamente nuova, senza dover reinstallare tutti i programmi da capo.

Dipartimento di Informatica e Scienze dell'Informazione

La struttura del filesystem

/var

Il filesystem var contiene dei file che cambiano, come le directory di spool (per la posta, le news, le stampanti eccetera), i file di log, le pagine man formattate e i file temporanei..

contiene i dati che vengono modificati quando il sistema lavora normalmente, è specifico per ciascun sistema, cioè non viene condiviso in rete con altri computer.

Dipartimento di Informatica e Scienze dell'Informazione

La struttura del filesystem

/home

Il filesystem /home contiene le home directory degli utenti, cioè tutti i dati sul sistema.

Separare le home directory su un albero o su un filesystem separato rende molto più semplici i backup: le altre parti in genere non hanno bisogno di backup, o almeno non frequentemente (cambiano poco nel tempo).

Una /home grande potrebbe dover essere separata in vari filesystem, cosa che richiede l'aggiunta di sottolivelli, ad esempio /home/students e /home/staff.

La struttura del filesystem

/bin

Contiene i comandi necessari durante il boot del sistema che possono anche essere usati dagli utenti normali dopo il boot.

/sbin

Come /bin, ma i comandi non sono intesi per gli utenti normali, anche se questi li possono usare se necessario e se hanno i permessi.

/root

La home directory dell'utente root.

La struttura del filesystem

/lib

Le librerie condivise necessarie ai programmi.

Molto importante e' /lib/modules che contiene i moduli del kernel caricabili, specialmente quelli che sono necessari per avviare il sistema quando lo si sta recuperando da un problema (ad esempio i driver di rete e dei filesystem).

/tmp

I file temporanei. I programmi che vengono avviati dopo il boot dovrebbero usare /var/tmp, non /tmp, dato che il primo si trova probabilmente in un disco con più spazio.

La struttura del filesystem

/boot

I file usati dal boot manager, come LILO. Le immagini del kernel spesso vengono tenute qui invece che nella directory root. Se ci sono diverse immagini di kernel, la directory può facilmente crescere parecchio, e spesso può essere meglio tenerla in un filesystem separato.

/mnt

Il punto di mount dove l'amministratore di sistema può montare temporaneamente delle directory. /mnt può essere diviso in sottodirectory (esempio /mnt/dosa può essere il floppy che usa un filesystem MS-DOS, e /mnt/exta lo stesso con un filesystem ext2)

La struttura del filesystem

/dev

La directory dev contiene degli speciali file di device, uno per ciascun dispositivo. I file di device vengono chiamati usando delle speciali convenzioni, che sono descritte nel'elenco dei dispositivi

I file di device vengono creati durante l'installazione, ma possono essere creati anche in seguito usando lo script chiamato MAKEDEV.

La struttura del filesystem

/proc

Il filesystem /proc contiene un filesystem virtuale:

In realtà non è per niente un filesystem, anche se gli somiglia. proc permette di avere facile accesso ad alcune strutture di dati del kernel, come la lista dei processi (da cui il nome). Fa apparire queste strutture di dati come un filesystem, e quel filesystem può essere manipolato con i normali strumenti di manipolazione di file.