

A.A. 08/09

Fondamenti di Programmazione

(canale E-O)

Docente: Prof.ssa Tiziana Calamoneri
calamo@di.uniroma1.it

Esercitatore: Dott. Roberto Petroccia
petroccia@di.uniroma1.it

Outline

- ✓ Introduzione a Linux
 - Cosa è Linux
 - Caratteristiche
 - Livelli
 - Software open source
 - Primi comandi Unix

Cosa è Linux

- È un SO
- Tesi di laurea di Linus Torwalds all'università di Helsinki in Finlandia
- Sviluppato a partire da Minix, un sistema UNIX molto piccolo
- 1991 è stata rilasciata la prima versione (0.02)
- Nel 1994 fu rilasciata la versione 1.0 del Kernel Linux
- ... lo sviluppo di Linux continua
- GNU General Public License: il suo codice sorgente è disponibile e gratuito
- È considerato un SO eccellente, economico alternativo ad altri SO molto costosi.

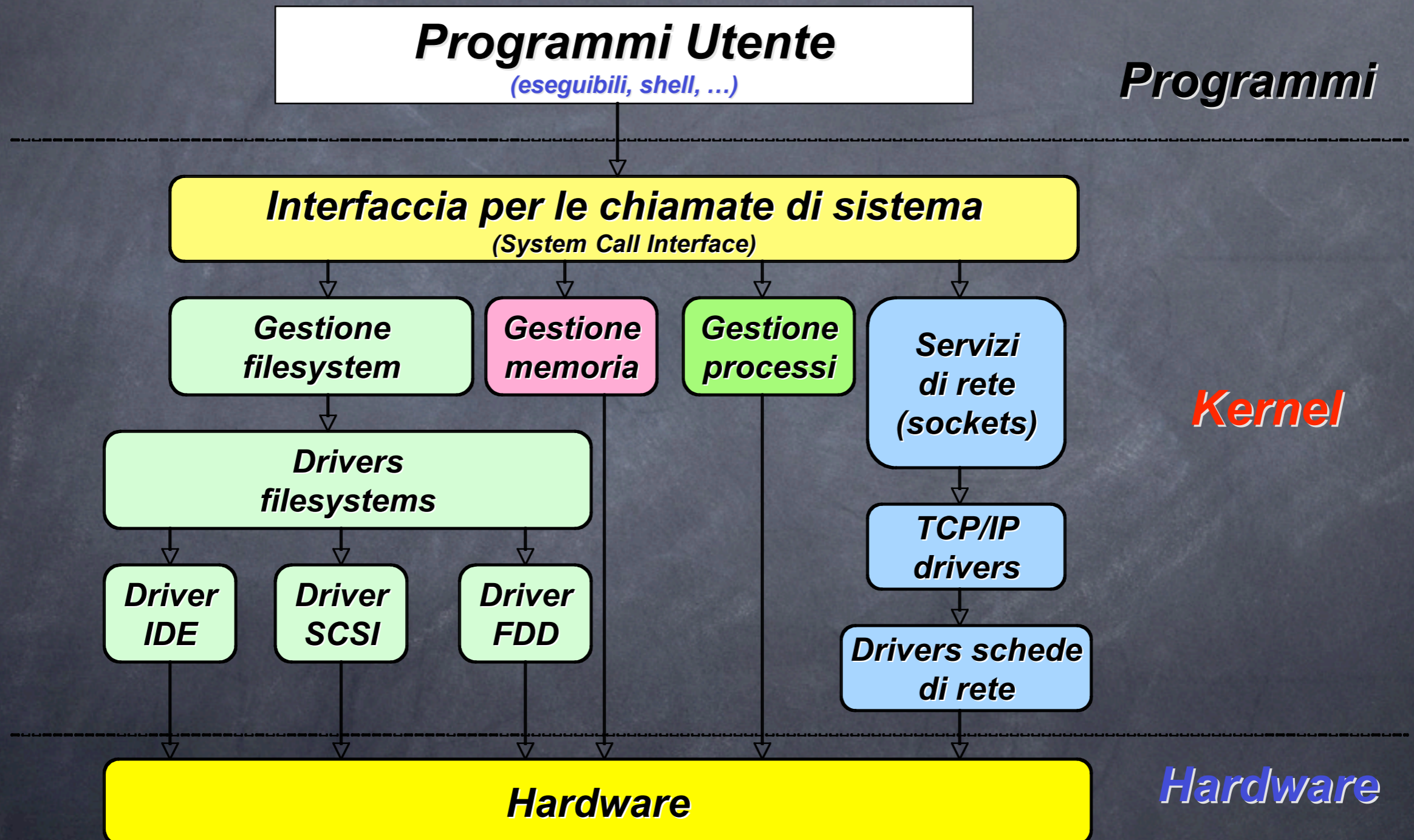
Caratteristiche

- Multitasking: più programmi funzionano contemporaneamente.
- Multiuser: più utenti nella stessa macchina contemporaneamente
- Scalabile: in modo testo ha pretese hardware minime
- Funziona su diverse architetture: PC, SPARC, Mac,...
- Affianca diversi SO: Windows, Solaris, MS-DOS, MacOs,...
- Riconosce parecchi filesystem comuni, ha una struttura di memorizzazione dati gerarchica
- Gestisce multiprocessor
- Architettura trasparente e invisibile all'utente

Caratteristiche

- Linux non usa estensione per i file e qualunque cosa viene trattata come un file.
- Accesso trasparente a partizioni MS-DOS (o partizioni OS/2 FAT), il tutto appare come un normale filesystem Unix (eccetto per ovvie restrizioni sui nomi di file, permessi e così via)
- Modalità protetta: ogni programma in esecuzione occupa specifici indirizzi di memoria, "protetti" dalla possibilità che altri programmi vadano ad occupare gli stessi indirizzi causando in tal modo il blocco del sistema
- Struttura a livelli collegati tra di loro
- Possibilità di utilizzare un completo ambiente grafico, una GUI (Graphical User Interface) conosciuta con il nome di X Windows
- Ambiente di sviluppo completo per applicazioni e programmi (i.e. il C ed il C++, il Perl il Tcl/Tk, etc. ...)

Architettura di Linux



Livelli

- **Livello Kernel:** nucleo del sistema operativo, gestisce direttamente la macchina dialogando fisicamente con l'hardware (memoria e processore)
- **Livello User:**
 - Livello dei comandi o processi che possono essere eseguiti contemporaneamente sul sistema. Tali comandi vengono eseguiti dal kernel che li passa direttamente al sistema
 - Livello della shell: interprete dei comandi (un po' come il COMMAND.COM del DOS), che permette all'utente di impartire i comandi che verranno poi eseguiti dal sistema, attraverso l'interazione del kernel. La shell rappresenta un ambiente di sviluppo completo

Software open source

Linux ha introdotto l'idea di OpenSource

- **Accesso al codice sorgente**
- **Redistribuzione gratuita del codice e dei programmi compilati**
- **Permesso di effettuare modifiche**
- **Preservazione dell'integrità del codice sorgente dell'autore**

Perché l'Open Source è importante

- **Apertura del codice**

Avere a disposizione il codice sorgente agevola la risoluzione di problemi

- **Stabilità**

Dal momento che l'OpenSource può contare su una grande comunità di sviluppatori e che tutto il codice sorgente sviluppato deve rimanere pubblico, il software Open Source è generalmente più stabile del software commerciale

- **Adattabilità**

Il codice OpenSource è strettamente collegato con delle regole standard (Open Standards) e quindi è facilmente adattabile ad ogni altra applicazione OpenSource

- **Qualità**

I prodotti OpenSource possono essere testati da una grande comunità, che è anche potenzialmente una comunità di sviluppatori

- **Innovazione**

Ciò che spinge l'OpenSource è la competizione e la possibilità di contribuire comunemente allo sviluppo di un prodotto

- **Costo**

La maggior parte del software OpenSource è gratis (ma non lo è necessariamente)

Cose importanti da ricordare

- Utilizzare l'utente root solo se necessario, altrimenti usate il vostro account personale
- Sfruttare il completamento automatico dei file con il tasto TAB
- Evitare di usare nomi contenenti spazi
- Per qualunque dubbio il web è una fonte inesauribile di risposte e la comunità Linux è molto attiva
- Potete trovare la bibbia sull'uso di linux "Appunti di informatica libera" all'indirizzo: <http://a2.pluto.it/a21.htm>
- Esiste un Linux User Group della Sapienza: <http://www.ls-lug.org/>

Primi comandi Unix --> ls

- Comando “**ls**”: restituisce la lista dei file e cartelle presenti nella directory corrente.
- Se digitiamo al prompt di una macchina linux “# ls -al” ci viene restituita una lista dei file.

Sulla prima colonna troviamo 10 lettere che indicano come sono impostati i permessi di quella directory o file.

Sulla seconda il numero di hard link (collegamenti fisici) (ma di questo ne parleremo un'altra volta.)

Sulla terza e quarta rispettivamente nome del proprietario e nome del gruppo.

Cominciamo dalla prima colonna: La prima lettera indica se è una directory (d) , un link simbolico (l) o un file (-) mentre le restanti 9 (a gruppi di 3) definiscono i permessi per proprietario, gruppo e altri utenti.

I tipi di permesso possono essere

1. lettura (r)
2. scrittura (w)
3. esecuzione (x)

Primi comandi Unix --> man

- Comando importantissimo “**man**”, è il manuale d’uso dei comandi
- Se digitiamo al prompt di una macchina linux “# man ls” otteniamo tutte le informazioni su come usare il comando “ls” più tutte le possibili opzioni da usare ed i comandi correlati
- Ogni qual volta non ricordiate come un comando debba essere usato o cosa faccia potete consultare il manuale (digitare man <comando>)

Primi comandi Unix --> cd

- Il comando “cd” viene usato per muoversi all’interno delle cartelle
- Il “.” rappresenta la directory corrente, digitando “cd .” si resta nella stessa directory
- Il “..” rappresenta la directory superiore alla vostra, digitando “cd ..” salite di un livello
- Il “/” rappresenta la directory di root, radice di tutto il filesystem
- Se digitiamo al prompt di una macchina linux “# cd <nomeDirectory>”, entriamo nella directory desiderata, invece del semplice nome potete usare anche interi percorsi, ad esempio:
“# cd /Users/Petrox/homework”

Primi comandi Unix --> mkdir

- Il comando “**mkdir**” viene usato per creare una nuova directory (sta per make directory)
- Se digitiamo “# mkdir <nomeDirectory>”, creiamo una nuova directory (chiamata <nomeDirectory>) all’interno della directory corrente, ovviamente bisogna avere i permessi giusti per scrivere in una directory.
- Potete usare anche interi path invece che il solo nome della directory, “# cd /Users/Petrox/homework” creerà la directory homework all’interno della directory Petrox. Quando usate i path dovete assicurarvi che le cartelle intermedie esistano e siano da voi accessibili

Primi comandi Unix --> cp

- Il comando “cp” viene usato per copiare file e directory
- La sintassi del comando è “cp <nomeFile> <directoryDestinazione>”
- Se digitiamo “# cp pippo /Users/Petrox”, il file pippo dalla directory corrente verrà copiato nella directory Petrox
- Se vogliamo copiare directory allora è utile usare l’opzione “-r” che effettua una copia ricorsiva ossia tutto il contenuto della directory verrà ricorsivamente copiato. Con “# cp -r pippo /Users/Petrox” la cartella pippo e tutto il suo contenuto sono copiati nella cartella Petrox

Primi comandi Unix --> mv

- Il comando “mv” viene usato per spostare o rinominare file e directory
- La sintassi del comando è “mv <nomeFile> <directoryDestinazione>” se vogliamo spostare un file
- La sintassi del comando è “mv <vecchioNome> <nuovoNome>” se vogliamo rinominare un file
- Se digitiamo “# mv pippo /Users/Petrox”, il file pippo dalla directory corrente verrà spostato nella directory Petrox
- Se digitiamo “# mv pippo pippo2”, il file pippo nella directory corrente verrà rinominato in pippo2

Primi comandi Unix --> rm

- Il comando “**rm**” viene usato per eliminare file e directory
- La sintassi del comando è “**rm <nomeFile>**”
- Per eliminare una directory bisogna usare l’opzione “**-r**”, ad esempio: “**rm -r <nomeDirectory>**”
- Fate attenzione quando eliminate i file per essere sicuri che stiate eliminando il file corretto.

Primi comandi Unix --> gcc

- Il comando “gcc” invoca il vostro compilatore, che è contenuto nel pacchetto di sviluppo di Linux
- gcc non è un semplice compilatore ma nasce da un grosso progetto.
- La sintassi del comando è “gcc <nomeFile>”, se chiamiamo “gcc pippo.c” ed il file pippo.c non contiene errori di sintassi viene creato un file “a.out” che è l’eseguibile. Per lanciare il programma basta poi digitare “./a.out”
- Esistono tantissime opzioni come ad esempio “-o” per specificare il nome del file di output, se chiamiamo “gcc pippo.c -o pippo.out” al termine della compilazione invece di a.out viene creato il file pippo.out. Per lanciare il programma basta poi digitare “./pippo.out”
- Esistono tantissime opzioni che potete usare, controllatele utilizzando il comando man
- Imparate a leggere l’output del compilatore in caso di errori, non basta dire c’è un errore ma è importante capire il perché