



Sistemi Operativi

2° Modulo

Anno Accademico 2010-2011

Prof. Claudio Cilli

Obiettivi del Corso

- Fornire le nozioni necessarie per utilizzare un sistema Linux attraverso i principali comandi di base, interagendo con l'interfaccia utente basata sulla linea di comando (shell)
- Insegnare le basi per la programmazione in linguaggio script (bash)
- Illustrare l'interfaccia applicativa (Application Programming Interface - API) che il SO offre al programmatore e l'utilizzo delle chiamate al sistema
- Chiarire mediante esempi ed esercizi alcuni concetti duali fondamentali, quali ad esempio:
 - applicazione utente / kernel
 - libreria C standard / API del SO (system calls)
 - memoria di processo statica / dinamica
 - compilatore / interprete
 - libreria statica / dinamica
 - codice sorgente / modulo oggetto
 - programma / processo

Prerequisiti e relazione con altri corsi

- E' richiesta la padronanza del linguaggio C
- Prevede che gli studenti seguano contemporaneamente (o abbiano seguito) il primo modulo del corso di Sistemi Operativi e che abbiano acquisito da esso i concetti fondamentali sui SO
- Il modulo si focalizzerà sui servizi base del SO per programmi sequenziali che interagiscono tra loro mediante filesystem o mediante semplici meccanismi di comunicazione tra processi (IPC), quali segnali e pipe
- Si farà riferimento principalmente all'ambiente Linux

Testi per il corso - I

- Nessun testo ufficiale adottato: in generale le slide e la documentazione in linea (comandi man, info) sono più che sufficienti
- Per una trattazione sistematica e approfondita dei concetti relativi ai sistemi operativi, si può fare riferimento ai libri di testo consigliati per il primo modulo di SO
- Segnaliamo alcuni testi/documenti per eventuali approfondimenti o chiarimenti sui vari argomenti (nel sito web del corso trovate i link alle versioni reperibili su Internet e ulteriori link ad utili tutorial correlati)
- Linux
 - E. Siever, A. Weber, S. Figgins, Linux in a Nutshell, Fourth Edition, O'Reilly
 - M. Garrels, Introduction to Linux - A Hands on Guide, May 2005
 - L. Wirzenius, J. Oja, S. Stafford, A. Weeks, The Linux System Administrator's Guide
 - A. Lissot, K. Koehntopp, Linux Partition HOWTO
 - E. S. Raymond, The Linux Installation HOWTO

Testi per il corso - II

- **Linguaggio C**
 - B. W. Kernighan, D. M. Ritchie, The C Programming Language (2nd Edition), Prentice Hall

- **Bash scripting**
 - Mike G, BASH Programming - Introduction HOW-TO, July 2000
 - M. Garrels, Bash Guide for Beginners, March 2005
 - M. Cooper, Advanced Bash-Scripting Guide, June 2005

- **Programmi eseguibili**
 - David Wheeler, Program Library HOWTO , April 2003
 - Hongjiu Lu, ELF: from the Programmer's Perspective
 - M. L. Haungs, Executable and Linking Format (ELF)

- **Editor Vi**
 - Vimbook-OPL
 - Bram Moolenaar, VIM USER MANUAL

Testi per il corso - III

- **Compilatore GCC**
 - R. Stallman, GCC Developer Community, Using the GNU Compiler Collection (GCC)
- **Debugger GDB**
 - Debugging with GDB - The GNU Source-Level Debugger, Free Software Foundation
- **binutils**
 - R. H. Pesch, J. M. Osier, The GNU Binary Utilities, Cygnus Support
- **Make**
 - R. M. Stallman, R. McGrath, P. Smith, GNU Make - A Program for Directing Recompilation
- **Programmazione di Sistema**
 - Richard Stevens, "Advanced Programming in the UNIX Environment", Addison-Wesley
 - M. Mitchell, J. Oldham, A. Samuel, "Advanced Linux Programming", New Riders Publishing, First Edition, June 2001

Nota: per ogni guida o manuale pubblicato sul Web, verificare di ottenere sempre la versione più recente

Classici da bibliografia

○ Sistemi Operativi

- W. Stallings, “Operating Systems: Internals and Design Principles”, Fifth Edition
- Silberschatz, P.B. Gavin, “Operating System Concepts”, Addison-Wesley
- A.S. Tanenbaum, “Operating Systems - Design and Implementation”, Prentice-Hall
- H.M. Deitel, “Operating Systems”, Addison-Wesley
- G. Nutt, “Operating Systems: a Modern Perspective”, Addison-Wesley

○ Tecniche di Programmazione

- B. W. Kernighan, R. Pike, “The Practice of Programming”, Addison-Wesley
- Eric Steven Raymond, “The Art of Unix Programming”, Addison-Wesley

Sito Web

- Il corso è dotato di un sito web:
 - http://twiki.di.uniroma1.it/twiki/view/Lab_so_1
- Esiste inoltre una mailing list
 - <https://lists.inroma.roma.it/mailman/listinfo/sistemioperativi>
- Troverete:
 - Obiettivi del corso
 - Programma del corso
 - Slide presentate a lezione
 - Esempi ed esercizi
 - Link alla documentazione
 - Modalità e compiti d'esame

Orari di ricevimento

- Ricevimento di persona:
 - al termine delle lezioni o nell'intervallo
 - nell'orario di ricevimento pubblicato sul sito del Dipartimento
 - per appuntamento previa richiesta via e-mail
- Ricevimento alternativo:
 - via e-mail tramite la lista Sistemi Operativi

Argomenti del corso

- Introduzione all'uso di Linux
- La Shell
- Programmi eseguibili
- Introduzione alle system call
- Allocazione di memoria
- Gestione dei processi
- I/O system call
- Inter Process Communication (IPC)
- Linee guida per la programmazione

Modalità di esame - I

- L'esame di Sistemi Operativi è unico (12 CFU) e diviso in due moduli
- **Prima possibilità**
 - una prova di esonero: un questionario a risposte multiple e uno a risposte aperte da svolgere in aula al termine del primo modulo
 - e
 - un progetto da sviluppare sul proprio PC a casa o utilizzando i PC del laboratorio di calcolo e una prova orale sugli argomenti del secondo modulo da sostenere nella data dell'appello

N.B: L'ambiente tecnico di riferimento è quello dei PC del laboratorio (livello di compilatore e librerie, versione della shell, ecc.). Gli studenti sono tenuti ad accertarsi che il loro compito funzioni correttamente in tale ambiente

Modalità di esame - II

- **Seconda possibilità**
 - una prova orale sugli argomenti del primo modulo e, contestualmente,
 - un progetto da sviluppare sul proprio PC a casa o utilizzando i PC del laboratorio di calcolo e una prova orale sugli argomenti del secondo modulo da sostenere nella data dell'appello

N.B: Il progetto è assegnato all'inizio delle lezioni del secondo modulo e potrà essere discusso entro un anno (il progetto assegnato nel 2011 sarà valido fino alla sessione di febbraio del 2012)

Modalità di esame - III

- Per alcuni studenti l'esame di Sistemi Operativi è diviso in due moduli indipendenti da 6 CFU (ovvero altre situazioni analoghe, come ad esempio il progetto INCOS). In questo caso:
 - una prova scritta: un questionario a risposte multiple e uno a risposte aperte da svolgere in aula per il **primo modulo**, eventualmente integrata da una discussione orale da sostenere nella data prevista per l'appello del primo modulo
 - un progetto da sviluppare sul proprio PC a casa o utilizzando i PC del laboratorio di calcolo e una prova orale sugli argomenti del **secondo modulo** da sostenere nella data dell'appello del secondo modulo

N.B: Il progetto è assegnato all'inizio delle lezioni del secondo modulo e potrà essere discusso entro un anno (il progetto assegnato nel 2011 sarà valido fino alla sessione di febbraio del 2012)

Regole valide per il progetto

- Il progetto può essere svolto individualmente o in gruppo (max 3 persone)
- Una volta consegnato il progetto è obbligatorio presentarsi all'appello immediatamente successivo per sostenere l'esame. I termini per la consegna saranno comunicati durante le lezioni
- Il progetto deve costituire una creazione originale, quindi non è possibile condividere parti di codice o della eventuale relazione, se richiesta, con altri studenti/gruppi, o copiare contenuti derivanti da altre fonti
- Le discussioni tra studenti, gli scambi di idee, l'utilizzo della mailing list di sistemi operativi sistemioperativi@inroma.roma.it ed in genere tutto ciò che aiuta lo studente ad apprendere, è invece legittimo ed apprezzato (in caso di dubbi riguardo a quali forme di collaborazione siano considerate legittime o meno, meglio chiedere esplicitamente chiarimenti via mail)
- Su esplicita richiesta dello studente, e previo assenso del docente, il risultato di uno scritto potrebbe essere modificato all'esame, mediante un'interrogazione supplementare: in questo caso, però, si potrebbe anche rischiare di peggiorare rispetto al voto dello scritto, a causa di un'interrogazione con esito negativo

Regole specifiche per il progetto

- Il progetto deve essere consegnato secondo le scadenze indicate
- Il progetto consiste in un programma in linguaggio C che soddisfi i requisiti specificati, utilizzando le chiamate di sistema (system call) che fanno parte del programma del corso. E' sconsigliato il ricorso ad altre system call
- L'esame orale consiste in una discussione sulle scelte progettuali e sull'implementazione del software. Prendendo spunto dal lavoro del progetto, potranno essere poste domande su vari argomenti facenti parte del programma del corso

Iscrizioni

- Per poter sostenere le prove è necessario:
 - attendere la comunicazione del compito a lezione o sul sito web
 - iscriversi individualmente o come gruppo (max 3 persone) via email al docente comunicando:
 - matricola, cognome e nome degli studenti componenti il gruppo di lavoro (uno studente per riga)
- Il docente assegna un identificativo ad ogni gruppo
- Consegnare i progetti entro le scadenze fissate (comunicate a lezione e tramite il sito web)
- Prenotarsi secondo le modalità indicate

Criteri di valutazione dei compiti (esoneri e progetti)

- Correttezza del codice: principale elemento di valutazione che determina (da solo!) il superamento dell'esame
- Gestione degli errori: fa parte integrante della correttezza del codice!
- Modularità e leggibilità del codice: divisione in funzioni, commenti, ...
- Qualità della documentazione: manuale utente, relazione sul progetto

Infine...

- Frequentare le lezioni e assimilare quanto più possibile gli argomenti spiegati
- Fare domande di interesse comune durante le lezioni
- Provare da soli varianti degli esempi svolti a lezioni e fare gli esercizi proposti
- Installare Linux sul proprio PC o utilizzare PC del laboratorio di calcolo/aula di lezione
- Memorizzare gli idiomi illustrati negli esempi a lezione per i costrutti e le system call principali
- Colmare le proprie lacune/incertezze su Linux e linguaggio C