Laboratorio di Sistemi Operativi I
Anno Accademico 2006-2007

Introduzione al corso

Francesco Pedullà
(Tecnologie Informatiche)

Massimo Verola
(Informatica)

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

- è richiesta la padronanza del linguaggio C (corsi Programmazione 1 e 2)
- LSO1 prevede come prerequisiti Architettura 2 e Programmazione 2 (come SO1)
- LSO1 prevede che gli studenti seguano contemporaneamente (o abbiano seguito) il corso di SO1 e che abbiano acquisito da esso i concetti fondamentali sui SO
- LSO1 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
- LSO2 affronterà in modo più completo e sistematico tutta l'IPC e il multithreading
- LSO1 farà riferimento soltanto all'ambiente Linux
- SO2 tratterà L'API di Windows (Win32 API)

Testi per il corso - I

- Nessun testo ufficiale adottato: in generale i lucidi 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 l'esame di SO1
- 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*

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

Classici da bibliografia

Per ulteriori approfondimenti, i seguenti libri sono consigliati:

Sistemi Operativi:

- *W. Stallings*, “Operating Systems: Internals and Design Principles”, Fifth Edition
- *Silberschatz, P.B. Gavin*, “Operating System Conceps”, 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 e Programmazione di Sistema:

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

Sito Web

- Il corso è dotato di un sito web:

  http://twiki.dsi.uniroma1.it/twiki/view/Lab_so_1

- Troverete:
  - Informazioni generali sul corso
  - Slide presentate a lezione
  - Esempi ed esercizi
  - Link alla documentazione
  - Modalità e compiti d'esame
  - FAQ

Orari e organizzazione delle lezioni

Informatica:

- Mercoledì ore 16-18
- Venerdì ore 16-18
- Periodo lezioni: dal 27 Sett 2006 al 22 Dic 2006
- Date esami: da definire

Tecnologie Informatiche:

- Giovedì ore 16-18
- Venerdì ore 16-18
- Periodo lezioni: dal 28 Sett 2006 al 22 Dic 2006
- Date esami: da definire

Orari di ricevimento

- Ricevimento sincrono (di persona):
  - al termine delle lezioni

  oppure

  - per appuntamento previa richiesta via e-mail o telefono

- Ricevimento asincrono: via e-mail, sempre

  Francesco Pedullà: fpedulla@it.ibm.com

  Massimo Verola: massimo.verola@quadrics.it

Argomenti del corso

0. La programmazione
1. Introduzione all'uso di Linux
2. La Shell e cenni sull’editor vi
3. Programmazione in linguaggio script
4. Librerie e programmi eseguibili
5. Programmazione di sistema

Modulo 0

- **La programmazione**
  - La modularità e le interfacce
  - La scelta degli algoritmi e delle strutture di dati
  - La notazione e lo stile
  - Il debugging e il collaudo
  - Appendice: Regole per la progettazione e implementazione del SW

Modulo 1

- **Introduzione all'uso di Linux**
  - Caratteristiche e struttura di Linux
  - Interfaccia con l'utente
  - Comandi base
  - Filesystem
  - Gestione e monitoraggio di processi

Modulo 2

- **Shell**
  - Tipi di shell
  - Comandi interni
  - Lancio di processi in background / foreground
  - Redirezione I/O e pipe
  - Comandi utili di sistema
  - Espressioni regolari
  - Appendice: Cenni sull'editor vi e su sed

Modulo 3

- **Programmazione in linguaggio script**
  - Variabili ed espressioni
  - Condizioni
  - Strutture di controllo: cicli e branch
  - Funzioni e passaggio di parametri
  - File di configurazione
  - Comandi built-in

Modulo 4

- **Programmi eseguibili**
  - Programmi e processi
  - Ciclo di vita di un programma
  - Stati di un processo
  - Compilatore, linker e loader
  - Formato ELF e binutils
  - Librerie statiche e dinamiche
  - Debugger
  - Make

Modulo 5

- **Programmazione di sistema**
  - Introduzione alle system calls
  - Allocazione della memoria
  - Creazione e controllo di processi
  - I/O system calls
  - IPC: segnali, pipe, FIFO

Modalità di esame - I

L'esame puo' essere superato in due diversi modi:

- ***due prove di esonero*** da sviluppare sul proprio PC a casa o utilizzando i PC del laboratorio di calcolo durante il periodo delle lezioni

- ***un progetto*** da sviluppare sul proprio PC a casa o utilizzando i PC del laboratorio di calcolo e ***una prova orale*** da sostenere nella data dell'appello

N.B: L'ambiente tecnico di riferimento e' quello dei PC del laboratorio (livello di compilatore e librerie, versione della shell, etc.). Gli studenti sono tenuti ad accertarsi che il loro compito funzioni correttamente in tale ambiente. Se il compito non funziona, ben difficilmente l'esame potra' essere superato!

Regole valide sia per gli esoneri che per il progetto:

- Sia gli esoneri che il progetto possono essere svolti i***ndividualmente*** o ***in gruppo (max 3 persone)***.
- Essi devono 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, il risultato di uno scritto (esonero o progetto) 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 gli esoneri (I):

- I compiti di esonero devono essere consegnati entro le scadenze indicate sul sito web del corso.
- L'esonero n.1 riguarderà lo scripting in bash shell, mentre l'esonero n.2 le system call.
- Ai fini del superamento dell'esame è necessaria la sufficienza in entrambi gli esoneri.
- Il voto finale è dato da una media pesata:
  1/4 * [voto esonero n.1] + 3/4 * [voto esonero n.2].
- Se si è superato solo l'esonero n.1 (bash scripting), prima di accedere all'orale, si deve svolgere il progetto. In sede di appello, si è esonerati dal sostenere domande sulla parte del corso riguardante la bash shell e la sua programmazione.
- Se si è superato solo l'esonero n.2 (system call), si può accedere direttamente all'orale nella data dell'appello per sostenere un'interrogazione integrativa sulle parti del corso non coperte dall'esonero n.2.

Regole specifiche per gli esoneri (II):

- In sede di appello d'esame, anche nel caso in cui non si debba sostenere un'interrogazione orale in quanto esonerati, è comunque possibile che venga chiesto al candidato di spiegare e commentare i programmi realizzati, per verificare che non sia stato copiato il lavoro di altri. Questo vuol dire che ogni studente, anche se ha lavorato in gruppo, deve dimostrare di conoscere e controllare dettagliatamente tutto il lavoro svolto (compiti di esonero o progetto), e non soltanto la parte da lui progettata, implementata e documentata.
- I voti degli esoneri vengono pubblicati sul sito web del corso.
- I voti degli esoneri rimangono validi fino all'ultimo appello della sessione di Settembre dell'anno accademico in corso, dopodiché bisogna svolgere l'esame secondo la modalità standard (progetto+orale).

Regole specifiche per il progetto (I):

- Il progetto deve essere consegnato secondo le scadenze indicate sul sito web del corso.
- 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.

Regole specifiche per il progetto (II):

- Una volta consegnato un progetto e ottenuto un voto sufficiente, e' altamente consigliabile presentarsi all'appello immediatamente successivo per sostenere l'orale. Ad ogni modo il voto del progetto verra' mantenuto valido per circa 6 mesi. In particolare, i progetti relativi agli appelli della sessione di Gennaio/Febbraio rimangono validi per tutta la sessione di Giugno/Luglio, quelli di Giugno/Luglio e quelli di Settembre fino alla sessione di Gennnaio/Febbraio dell'anno successivo.
- Se non si riesce a superare l'esame, di regola non ci si puo' presentare agli appelli della stessa sessione. Deroghe a tale regola dipenderanno da situazioni specifiche e saranno comunicate dal docente al diretto interessato.

Iscrizioni

Per poter sostenere le prove è necessario:

- attendere la comunicazione del compito a lezione (esoneri) o sul sito web (progetto)
- 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 tramite l'apposita pagina web almeno 5 giorni prima dell'esame, per permettere di valutare la durata di ogni sessione

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
- **Numero dei componenti del gruppo:** gli errori, le imperfezioni e le incompletezze del compito hanno un peso crescente all'aumentare del numero dei componenti del gruppo. In generale, il voto risulta inferiore di 2-3 punti per ogni componente aggiuntivo.

Manuale d'uso del corso

- 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 farsi dare un'utenza sui PC del laboratorio di calcolo
- memorizzare gli idiomi illustrati negli esempi a lezione per i costrutti e le system call principali
- colmare sa soli le proprie lacune/incertezze sul linguaggio C

Con questi accorgimenti l'esame si riduce al progetto vero e proprio, e non allo studio di nozioni.