

Programma Dettagliato

- Comandi Linux (i comandi più importanti sono riportati in grassetto):
 - reperire informazioni: **man**
 - file e directory: **ls** (-l -a), **cd**, **pwd**, **cp**, **mv**, **ln** (-s), **mkdir**, **rmdir**, **rm**, **touch**
 - permessi: **chmod**, **chown**, **chgrp**
 - BASH: caratteri jolly
 - BASH: redirectione e pipe
 - BASH: command substitution e arithmetic substitution
 - manipolare file di testo: **cat**, **sort** (-n -k), **uniq**, **grep** (-v), **sed** (solo il comando s), **awk** (tutto, tranne cicli, salti condizionati ed array), **wc** (-c -w -l)
 - processi: **jobs**, **fg**, **bg**, **ps** (a u x), **kill**
 - altri comandi che è utile conoscere: **emacs**, **vi**, **pico**, **diff**, **type**, **file**, **less**, **head**, **tail**, **clear**, **sudo**, **passwd**, **lpr**, **groups**, **id**, **find**, **ssh**
- Programmazione di Shell BASH:
 - variabili
 - passare argomenti agli script
 - strutture di controllo: **if**, **for**, **while**, **&&**, **||**
 - comandi per script: **echo**, **exit**, **break**, **read**, **test** (operatore []).
- Programmazione di Sistema UNIX:
 - il debugger **gdb**
 - input/output di basso livello

- funzioni fondamentali: open, close, read, write, lseek, stat, dup, dup2
- altre funzioni: chmod, chown, link, unlink, symlink, readlink, mkdir, rmdir, chdir
- funzioni di lettura directory: opendir, readdir, closedir
- segnali ed allarmi: signal, kill, alarm
- gestione dei processi: sleep, fork, getpid, getppid, wait, waitpid, exec
- le pipe
- i thread POSIX
 - creazione, terminazione, attributi di un thread, i thread e fork/exec, i thread e i segnali
- sincronizzazione tra thread: mutex e condition variable
- thread-specific data
- programmazione di rete: i socket locali e TCP
 - system call: socket, bind, listen, accept, connect, get/setsockopt
 - funzioni di conversione del byte order
 - funzioni di manipolazione di indirizzi IP: gethostbyname, inet_aton, inet_ntoa
 - i server concorrenti
- comandi di supporto alla programmazione: strace, lsof, top, hexdump
- comandi di supporto alla programmazione di rete: netstat, ping, telnet

Modalità d'esame

L'esame consta delle seguenti parti:

- Una prova scritta relativa alla prima parte del corso (ex modulo A). La prima parte del corso termina con l'argomento “segnali ed allarmi”.
- Lo svolgimento di un progetto di gruppo e la stesura di una corrispondente relazione. I gruppi comprendono al più tre elementi.
- Un esame orale che consiste nella discussione del progetto, ma che può anche includere domande relative all'intero programma.

Gli studenti/esse devono concordare la traccia del progetto con il docente. Di norma, un gruppo di studenti chiede un progetto inviando un messaggio e-mail al docente, in cui sono specificati i componenti del gruppo.

Per prima cosa, un gruppo deve consegnare il progetto e la relativa relazione. A quel punto, tutti i membri del gruppo possono sostenere la prova scritta. Indipendentemente dal fatto che superino o meno la prova scritta, tutti i componenti del gruppo devono sostenere insieme la discussione del progetto. Ciascun membro del gruppo può scegliere di sostenere la prova scritta in un appello diverso,

purché il progetto sia stato consegnato, e non oltre la fine dell'anno accademico (marzo 2009). La relazione va consegnata in formato cartaceo (in assenza del docente, la relazione può essere depositata presso la casella di posta del docente nell'edificio Ma), mentre il codice del progetto deve essere inviato al docente in un file compresso in formato zip o tar.

Nota: Coloro che, appartenendo al vecchio ordinamento, hanno già superato e registrato la prima parte del corso sono tenuti a svolgere il progetto e a sostenere l'esame orale, che verterà solo sulla seconda parte del corso.

