

# **Programma del corso di laboratorio di sistemi operativi**

## **Anno accademico 2010/2011**

Gennaro Oliva

### ***Prima parte: La programmazione di shell***

#### ***Introduzione sui sistemi UNIX***

Breve storia del sistema operativo UNIX, le caratteristiche fondamentali e l'architettura del sistema operativo, l'autenticazione, i gruppi di utenti, il file system, i pathname, i permessi di protezione del file system.

#### **La shell bash**

Shell, comandi, il manuale online, l'interazione con il file system, la visualizzazione di file, la manipolazione dei permessi, il file globbing, i canali di comunicazione standard, la redirectione, le pipeline, le variabili, le variabili d'ambiente, la variabile PATH, l'espansione della shell, il quoting, l'exit status, le liste di comandi.

#### **Gli script di shell**

L'esecuzione di file interpretati, gli argomenti sulla linea di comando, il comando exit, il costrutto if-then-else-elif-fi, il comando test, il costrutto case-esac, il costrutto for-do-done, il comando seq, i costrutti while-do e until-do, le espressioni aritmetiche, i comandi true, false, continue, break, shift, la variabile RANDOM, il comando read, l'espansione negli script, le funzioni di shell e lo scoping delle variabili, l'editor vi.

#### **Le espressioni regolari**

Il concetto di espressione regolare, il comando grep, le espressioni regolari di base ed estese, il comando sed.

#### **Il linguaggio awk**

Le caratteristiche di awk, la sintassi delle regole, le regole speciali BEGIN ed END, i record e i field, le variabili built-in, l'istruzione printf, i pattern, la gestione dei tipi, gli operatori aritmetici, le variabili e gli argomenti sulla linea di comando, la redirectione, i costrutti if-else, while, do-while, for, le istruzioni break, continue, exit, gli array associativi, le funzioni numeriche, le funzioni per la manipolazione delle stringhe.

### ***Seconda parte: programmazione UNIX***

#### ***Le system call e la documentazione online***

Caratteristiche generali delle system call, le sezioni del manuale on line, i paragrafi delle pagine di manuale.

#### **Primitive per l'Input/Output**

Politiche di buffering, canali I/O standard, open(2), close(2), l'offset di un file aperto,

lseek(2), file sparsi, read(2), write(2), errno(3), perror(3), la condivisione di file aperti, le process table, le file table, la v-node table, l'opzione O\_APPEND, dup(2), dup2(2).

## **Primitive per l'interazione con il filesystem**

Delayed write, sync(2), fsync(2), fdatasync(2), stat(2), fstat(2), lstat(2), la struct stat, tipi di file, real ed effective UID e GID, access(2), permessi di file e directory, partizioni UNIX, cylinder group, superblock, i-node map, block bitmap, i-node, data block, directory block, hard link, link count, link(2), unlink(2), stickybit, soft link, date memorizzate nella struct stat, user file creation mask (umask), gestione delle directory con mkdir(2), rmdir(2), opendir(3), readdir(3), la struct dirent e closedir(3).

## **I segnali**

Le caratteristiche dei segnali, le condizioni che generano segnali, la protezione e i segnali, la cattura di un segnale, signal(2), alarm(2), pause(2).

## **La gestione dei processi**

Il concetto di processo, l'esecuzione di un programma in ambiente UNIX, le aree di memoria di un processo, il PID, il processo init, i runlevel, fork(2), copy-on-write, l'ordine di schedulazione di genitore e figlio, la condivisione di file aperti tra genitore e figlio, le similitudini e le differenze tra genitore e figlio, la terminazione di un processo, gli exit handler, il ciclo di vita di un processo, lo stato zombie, wait(2), waitpid(2), le macro per l'interpretazione della terminazione di un processo, la famiglia exec.

## **Pipe e FIFO**

Caratteristiche e limiti delle pipe, pipe(2), le pipe nella comunicazione tra genitore e figlio, la lettura su pipe e la scrittura da pipe, le broken pipe, la redirectione dei canali standard con le pipe (la pipeline). Le caratteristiche delle FIFO, le broken FIFO, caratteristiche dell'I/O su FIFO, le applicazioni client-server mediante FIFO.

## **I socket**

Il modello client-server, gli indirizzi IP, le porte, il concetto di endpoint, socket(2), la codifica degli interi e le funzioni di conversione, INADDR\_ANY, bind(2), listen(2), accept(2), lo schema di un'applicazione server, connect(2), lo schema di un'applicazione client, l'I/O su socket, i server iterativi e concorrenti, i nomi simbolici, la risoluzione, il funzionamento del resolver, gethostbyname(3).

## **I thread POSIX**

Caratteristiche di un thread, le aree di memoria di un processo multi-thread, la condivisione delle risorse, la libreria pthread, la gestione degli errori, pthread\_create(3), il passaggio di parametri, la terminazione di un thread, il valore d'uscita di un thread, pthread\_join(3), gli attributi di un thread, il detached state, la cancellabilità, thread fork ed exec e segnali, pthread\_kill(3), le funzioni thread-safe, le funzioni reentrant, il problema della sincronizzazione tra thread, il caso della modifica di una variabile, le race condition, i mutex, pthread\_mutex\_lock(3), pthread\_mutex\_unlock(3), pthread\_mutex\_trylock(3), atomicità dei mutex, i deadlock, le condition variable, l'utilizzo della condition variable nel problema produttore-consumatore, i thread-specific data, la funzione distruttore, thread-specific errno, pthread\_key\_create(3), pthread\_setspecific(3), pthread\_getspecific(3).