

Università di Verona
Corso di Laurea in Informatica/Informatica Multimediale
Laboratorio di Sistemi Operativi
01 Luglio 2008

Esercizio 1 (16 punti)

Il comando UNIX “grep” ricerca nei file d’ingresso indicati (o in tutti i file contenuti in una cartella, qualora sia indicata una cartella) tutte le righe che contengono una corrispondenza con il *modello* specificato. Sintassi:

grep opzioni {modello} {elenco file o cartella}

Scrivere uno script bash che si basi sul comando **grep** per cercare in modo ricorsivo (opzione “-r”) tutti i file contenenti la propria matricola, partendo da una cartella precisa del file system. Lo script deve ricevere 4 argomenti nella linea di comando:

- La *matricola*, ovvero la stringa corrispondente alla matricola personale.
- La *cartella*, ovvero il percorso completo da “/” della cartella da cui fare la ricerca.
- Il *file_output.log*, ovvero il percorso completo dove salvare tutto ciò che il comando manda sullo standard **output**.
- Il *file_error.log*, ovvero il percorso completo dove salvare tutto ciò che il comando manda sullo standard **error**.

Lo script deve controllare che il numero di argomenti della linea di comando sia giusto ed in caso contrario deve visualizzare un messaggio indicante il corretto utilizzo dello script. Infine lo script deve visualizzare un messaggio di errore anche nel caso in cui la directory fornita come argomento alla linea di comando non esista.

Esercizio 2 (16 punti)

Scrivere un programma C per la gestione della ricerca ricorsiva di stringhe tramite **grep**.

Si crei un processo padre P che deve richiedere all’utente: la stringa di cui fare la ricerca (modello) e la directory da cui partire con la ricerca ricorsiva del modello.

Il processo P deve inoltre successivamente creare un secondo processo F (figlio) che esegue il comando **grep**.

Il processo F deve eseguire la ricerca ricorsiva della stringa a partire dalla cartella specificata tramite *grep* e stampare a video il risultato.

Il processo P attende la terminazione del figlio e termina il programma.

Eventuali System Call da utilizzare:

```
#include <stdio.h>                                #include <unistd.h>
#include <sys/wait.h>                              #include <sys/types.h>
int pipe(int filedes[2]);                          pid_t fork(void);          int kill(pid_t pid, int sig);
int execl(const char *path, const char *arg, ..., char * const envp[]);
int execlp(const char *path, const char *arg, ...);
int execlp(const char *file, const char *arg, ...);
int execv(const char *path, char *const argv[]);
int execvp(const char *file, char *const argv[]);
int dup2(int oldfd, int newfd);
ssize_t read(int fd, void *buf, size_t count);
ssize_t write(int fd, const void *buf, size_t count);
int close(int fd);                                int dup(int oldfd);
int open(const char *pathname, int flags, mode_t mode);
int creat(const char *pathname, mode_t mode);
sighandler_t signal(int signum, sighandler_t handler);
pid_t wait(int *status);                          pid_t waitpid(pid_t pid, int *status, int options);
```