Esame del 06/03/2023 (GitHub)

## Esercizio 1

*Utilizzando opportuni comandi in concatenazione si eseguano le seguenti operazioni:*

(a) *Utilizzando awk si scriva un comando che stampi una lista dei file presenti nella directory corrente mostrando solo nome e proprietario.*

```
ls -l | awk '{print $9" "$3}'
```

(b) *Si calcoli la dimensione occupata in totale dai file regolari con dimensione inferiore di 1024 byte nella directory corrente.*

```
ls -l | awk '{ if($1 ~ /^-/ && $5<1024) somma=somma+$5 } END {print somma}'
```

(c) *Dato un file "parole.txt" stampa solo le linee con più di 10 caratteri.*

```
awk --field-separator= '{if(NF>10) print $0}' parole.txt
```

(d) *Impostando una variabile d'ambiente LIST: perm, link, user, group, date (può anche essere inizializzata al di fuori del comando stesso) visualizzi il listato dei file nella directory corrente con il campo corrispondente*

```
export LIST="perm,link,user,group,date"
ls -l | awk -v list="$LIST" '{split(list, fields, ","); print $fields[1], $fields[2], $fields[3],
$fields[4], $fields[5]}'
```

## Esercizio 2

Si realizzi uno script di shell BASH "`menu`", che implementi le seguenti funzioni accessibili da un menu:

(a) Aggiungi verifica - Chiede all'utente gli elementi: giorno, mese, anno, nome studente, voto. E inserirli in un file di nome "`verifica`". Il file deve essere creato sola la prima volta, chiamate successive, dovranno aggiungere nuove righe allo stesso file.
(b) Conta - Chiede all'utente il mese e lo studente, e conta il numero di prove effettuate nel mese per lo studente dato.
(c) Media - Chiede all'utente lo studente, e calcola la media dei voti delle verifiche date dallo studente.

Si rappresentino i mesi con una stringa di tre caratteri (gen, feb, mar, ecc.)

```
#!/bin/bash

scelta=0
while [ $scelta -ne 4 ];
do
    echo "Scegli un'opzione"
    echo "1) Aggiungi verifica"
    echo "2) Conta"
    echo "3) Media"
    echo "4) Uscita"
    read scelta
    case $scelta in
        1) echo "Inserisci il giorno: "
           read giorno
           echo "Inserisci il mese nel formato di 3 lettere (gen feb mar): "
           read mese
           echo "Inserisci l'anno: "
           read anno
           echo "Inserisci il nome dello studente: "
           read nome
           echo "Inserisci il voto: "
           read voto
```

```
            echo $giorno,$mese,$anno,$nome,$voto >> verifica
            ;;
        2) echo "Inserisci il nome dello studente: "
            read nome
            echo "Inserisci il mese in cui contare le prove: "
            read mese
            risultato=$(awk -F, -v mese="$mese" -v nome="$nome" 'BEGIN{somma=0} {if($2==mese && $4==nome)
somma=somma+1} END {print somma}' verifica)
            echo "In $mese ci sono $risultato prove per $nome"
            ;;
        3) echo "Inserisci il nome dello studente: "
            read nome
            somma=$(awk -F, -v nome="$nome" 'BEGIN{somma=0} {if($4==nome) somma=somma+$5} END {print somma}'
verifica)
            numero=$(awk -F, -v nome="$nome" 'BEGIN{somma=0}{if($4==nome) somma=somma+1} END {print somma}'
verifica)
            media=$((somma/numero))
            echo "La media è: $media"
            ;;
        4) echo "Uscita"
            ;;
        *) echo "Scelta non valida"
            ;;
    esac
done
```

**Esercizio 3**

*Si scriva un programma in C che prende in input i seguenti valori: filein Comando Cstop Cecc dove:*

- *filein: nome di un file leggibile.*
- *Comando: nome di un file eseguibile.*
- *Cstop, Cecc: singoli caratteri.*

*Il processo iniziale (P0) deve creare un processo figlio (P1). P1 dovra' leggere il contenuto del file filein, e trasferirlo integralmente al processo padre P0. Il processo P0, una volta creato il processo figlio P1, dovrà leggere e stampare sullo standard output quanto inviatogli dal processo figlio P1, secondo le seguenti modalità:*

- *Ogni carattere letto diverso da Cstop e da Cecc, viene stampato da P0 sullo standard output;*
- *Nel caso in cui P0 legga il carattere Cstop, dovrà semplicemente terminare forzatamente l'esecuzione di entrambi i processi;*
- *Nel caso in cui P0 legga il carattere Cecc, P0 dovrà interrompere l'esecuzione del figlio P1; P1 dal momento dell'interruzione in poi, passerà ad eseguire il comando Comando, e successivamente terminerà.*

*Scegliere un comando semplice da eseguire, es. ls o pwd. Stampare a video i diversi comportamenti.*

```c
#include "stdio.h"
#include "stdlib.h"
#include "signal.h"
#include "unistd.h"

char* comando;

void signal_handler(int s){
    printf("\nP1: Ricevuto il segnale.");
    execlp(comando, comando, NULL);
    exit(0);
}

int main(int argc, char** argv){
    printf("\nInizio del programma\n");
    comando = argv[2];
    char Cstop = argv[3][0];
    char Cecc = argv[4][0];
    int comm_pipe[2];
```

```
    pipe(comm_pipe);

    int pid = fork();
    if(pid==0){
        signal(SIGUSR1, signal_handler);
        close(comm_pipe[0]);
        printf("P1: Inizio a leggere da file.\n");
        FILE* filein = fopen(argv[1], "r");
        char c;
        do{
            c = fgetc(filein);
            write(comm_pipe[1], &c, sizeof(char));
        } while (c!=EOF);
        printf("P1: Finito di leggere, aspetto il segnale.\n");
        pause();
    }

    close(comm_pipe[1]);
    printf("P0: Inizio a leggere da pipe.\n");
    char c;
    printf("P0: ");
    do{
        read(comm_pipe[0], &c, sizeof(char));
        printf("%c",c);
        if(c==Cstop) kill(pid, SIGKILL), exit(0);
        if(c==Cecc) kill(pid, SIGUSR1), exit(0);
        if(c==EOF) printf("P0: Nessun carattere significativo ricevuto.\n"), exit(1);
    } while(1);

}
```