Esame del 13/06/2023 (GitHub)

Esercizio 1

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

A. *Dato un file avente contenuti "nome cognome" restituire a video o in un altro file, i nomi e cognomi in maniera inversa e separati da virgola. Es. "alessandra rossi"-> "rossi, alessandra".*

```
awk '{ print $2", "$1 }' nomicognomiesercizio1.txt
```

B. *Usando awk stampare tutti i numeri di telefono presenti in un file avente struttura: nome, cognome, numero di telefono, indirizzo.*

```
awk -F , '{ print $3 }' nomicognomiesercizio1.txt
```

C. *Usando awk, dato un nome prestabilito, restituire il numero di telefono della persona indicata. Si usi lo stesso file usato al punto precedente.*

```
read -p "Inserisci il nome: "; $nome; awk -F , '{ if( $1 == "'$nome'") print "Il suo numero è: " $3 }' nomicognomiesercizio1.txt
```

D. *Data una file contente nomi, cognome e data di nascita, usando awk restituire tutti i cognomi che iniziano con R e termina con E.*

```
awk '$2 ~ /^R.*e,/' nomicognomiesercizio1.txt
```

Esercizio 2

Dato un file di testo "paghe.txt" con almeno 6 righe di testo, scrivere uno script "stipendi" che inserisca il titolo "Sig.re" se si tratta di un uomo, e il titolo "Sig.ra" se si tratta di una donna, prima del nome. Calcolare e mostrare a video lo stipendio minimo, massimo e medio del personale, e aggiungere un bonus di x euro (dove x viene inserito dall'utente) allo stipendio minimo. Il file dovrà contenere i seguenti campi: nome, cognome, genere, stipendio, anno di assunzione.

```
#!/bin/bash

aggiungi_titolo(){
    if [ "$1" == "M" ]; then
        echo "Sig.re $2"
    elif [ "$1" == "F" ]; then
        echo "Sign.ra $2"
    else
        echo $2
    fi
}

stipendio_minimo=9999999
stipendio_massimo=0
media_stipendi=0
numero_dipendenti=0
while IFS=, read -r  nome cognome genere stipendio assunzione; do
    nome_completo=$(aggiungi_titolo "$genere" "$cognome")
    if [ "$stipendio" -lt "$stipendio_minimo" ]; then
            stipendio_minimo="$stipendio"
    fi
    if [ "$stipendio" -gt "$stipendio_massimo" ]; then
        stipendio_massimo="$stipendio";
    fi
    media_stipendi=$(( media_stipendi + stipendio ))
    numero_dipendenti=$(( numero_dipendenti + 1 ))
echo "$nome_completo, $nome, $genere, $stipendio, $assunzione"
done < paghe.txt
echo "Stipendio massimo: $stipendio_massimo"
echo "Stipendio minimo: $stipendio_minimo"
media_stipendi=$(( media_stipendi / numero_dipendenti ))
echo "Media stipendi: $media_stipendi"
```

```
read -p "Inserisci il bonus per lo stipendio minimo: " bonus
read -p "Si vuole modificare il file incrementando lo stipendio minimo? (y/n) " modifica

if [ "$modifica" == "y" ]; then
    sed -i "s/$stipendio_minimo/$((stipendio_minimo+bonus))/g" paghe.txt
else
     sed "s/$stipendio_minimo/$((stipendio_minimo+bonus))/g" paghe.txt
fi
```

Esercizio 3

Si realizzi un programma C il cui processo padre P0 dia il via alla generazione di n processi in gerarchia lineare. Dove n è un numero intero passato come argomento al programma. Cioè, P0 genera P1, P1 genera P2, ..., Pn genera Pn+1. Il P0 deve prendere in input una sequenza di N comandi (per semplicità, senza argomenti e senza opzioni). Ogni N processo deve eseguire il rispettivo N. L'applicazione termina quando l'ultimo processo ha terminato.

```
#include "stdio.h"
#include "stdlib.h"
#include "unistd.h"
#include "sys/wait.h"


int main(int argc, char *argv[]) {

    printf("\nEsecuzione di %s\n\n", argv[0]);

    for(int i = 1; i < argc; i++) {
        int pid = fork();
        if(pid == 0) {
            execlp(argv[i], argv[i], NULL);
            printf("Errore nell'esecuzione di %s\n", argv[i]);
            exit(1);
        }
    }

    while (wait(NULL) > 0);


    printf("Esecuzione terminata\n");
    return 0;
}
```