Esame del 13/06/2023 ([GitHub](https://github.com/davide-dipierro/Prove-svolte-LSO/tree/47190ac332ebc9af9ad26a14c280c89ac45304de/13062023_B))

**Esercizio 1**

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

1. *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

1. *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

1. *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

1. *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 scriva un programma in C che realizza un meccanismo di mailbox tra due processi. Le caselle di posta dei due processi sono costituite da due file "mail1.txt" e "mail2.txt" che si riferiscono rispettivamente al processo P1 e P2. Il processo P1 legge una stringa di 6 caratteri da tastiera (verificare se sono di meno o di più), mette la stringa nella mailbox del P2, e avvisa P2 mediante un segnale SIGUSR1. P2 legge la posta, e lo inserisce in maniera inversa (es. parola-> alorap) nella mailbox di P1.*

#include "stdio.h"

#include "stdlib.h"

#include "unistd.h"

#include "signal.h"

#define buffer\_size 6

void inverti(char\* buffer){

    for(int i = 0; i < buffer\_size/2; i++) {

        char temp = buffer[i];

        buffer[i] = buffer[buffer\_size - i - 1];

        buffer[buffer\_size - i - 1] = temp;

    }

}

void signal\_handler(int signo){

    ;

}

int main(int argc, char\*\* argv){

    int pid = fork();

    if(pid == 0){ // Codice P1

        char buffer[buffer\_size];

        printf("P1: Inserisci il messaggio: ");

        scanf("%s", buffer);

        FILE\* file = fopen("mail2.txt", "w");

        fwrite(buffer, sizeof(char), buffer\_size, file);

        fclose(file);

        kill(getppid(), SIGUSR1);

        printf("P1: Ho inviato il messaggio\n");

        signal(SIGUSR1, signal\_handler);

        pause();

        printf("P1: Ho ricevuto il segnale\n");

        FILE\* file2 = fopen("mail1.txt", "r");

        fread(buffer, sizeof(char), buffer\_size, file2);

        fclose(file2);

        printf("P1: Ho letto il messaggio: %s\n", buffer);

        return 0;

    }else{ // Codice P2

        signal(SIGUSR1, signal\_handler);

        pause();

        printf("P2: Ho ricevuto il segnale\n");

        char buffer[buffer\_size];

        FILE\* file2 = fopen("mail2.txt", "r");

        fread(buffer, sizeof(char), buffer\_size, file2);

        fclose(file2);

        printf("P2: Ho letto il messaggio: %s\n", buffer);

        inverti(buffer);

        printf("P2: Ho invertito il messaggio\n");

        FILE\* file3 = fopen("mail1.txt", "w");

        fwrite(buffer, sizeof(char), buffer\_size, file3);

        fclose(file3);

        kill(pid, SIGUSR1);

        printf("P2: Ho inviato il messaggio\n");

        return 0;

    }

}