

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 contenente 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 "Sig.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=$(( numero_dipendenti + 1 ))
done < paghe.txt
echo "$nome_completo, $nome, $genere, $stipendio, $assunzione"
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;
}
```