

# Progetto: Scheda Anagrafica

Muhammed Refati — VR458240

14 dicembre 2021

## 1 Introduzione

Il codice inizia introducendo la variabile `COLUMNS`, che memorizza il valore di `tputs cols`, ovvero una funzione implementata in `bash` che permette di poter stampare il numero di colonne del terminale. Definisco successivamente la variabile `scd`, a cui passo la stringa da implementare nell'output ed utilizzo la funzione `printf` per poter stampare tale stringa. Nel campo della variabile della stringa passo il valore di `scd` come stringa, utilizzando `${#scd}` e accompagnato dal valore delle colonne, cioè `COLUMNS/2`. Questa operazione permette di poter stampare la stringa nel centro nel terminale, indipendentemente dalla misura del terminale stesso. Chiamo successivamente la funzione per la stampa del menu a cui passo il valore di `$ filename`. La variabile `$ filename` memorizza `$ 1`, ovvero il path del file che l'utente inserisce da terminale. Se non si dovesse inserire alcun path, lo script copia il file pre-impostato degli studenti (`filestudenti.csv`) in una directory che crea sulla Scrivania e passa alla variabile `filename` tale percorso. Il controllo avviene tramite il comando `if` con il flag `-z`, che restituisce un valore booleano a seconda o meno dell'esistenza del file. Per questo motivo è importante far eseguire lo script nella directory in cui si scompatta l'archivio `zip`.

Per poter eseguire lo script bisogna prima eseguire il comando `"chmod 755 [nomedelfile]"` per poter dare i permessi di lettura ed esecuzione; successivamente lo si fa partire con `./[nomedelfile]` e il path del file (come, ad esempio, `/tmp/studenti.txt` o qualsiasi path in cui si trova il file da analizzare).

## 2 Funzione "menu", "choice" e "print"

### 2.1 Funzione menu

Il codice prosegue chiamando la funzione `menu`, che è una funzione di stampa che permette allo user di poter visualizzare le scelte possibili. La funzione `read` implementata in `bash` legge la variabile inserita in input e passa il suo valore alla funzione `choice` (`choice $ scelta`).

### 2.2 Funzione choice

La funzione `choice` implementa un case che prende in input la variabile passata dalla funzione `menu` (`$ 1`) e controlla che tale valore sia presente nel case e, a seconda della scelta effettuata, richiama le funzioni assegnate ad ogni valore. Nel caso venga inserito un valore non presente nel menu di scelta, la funzione richiama di nuovo il menu, per poter leggere il nuovo valore.

Al termine del case viene inserita la funzione `print`, che è una funzione di stampa (che verrà analizzata nel prossimo punto). Il punto 3 del case, ovvero la stampa del file degli studenti ordinata il file per numero di matricola crescente e, inoltre, presenta la funzione `tput setaf` implementata in `bash`, che permette di poter colorare il test con colori estesi fino a 256; tale funzione prevede di essere chiusa tramite `tput sgr0`, che imposta il valore del colore a quello standard. Si utilizza, inoltre, la funzione `column` che ordina la stringa `"NOME;COGNOME;MATRICOLA;ANNO"` e il file degli studenti inserito, sfruttando il flag `-t` e `-s";"`; il flag `-s";"` determina una tabulazione (`-t`) a livello del carattere `";"`, ovvero sostituisce `";"` con il tab.

### 2.3 Funzione print

La funzione `print` permette allo user di poter uscire dal programma o di poter ritornare nel menu di scelta. Tale funzione viene utilizzata per poter permettere l'uscita dal case e svuotare lo stack che la funzione precedente aveva riempito (anche tramite ricorsione). Al fine di effettuare un controllo più preciso sullo svuotamento dello stack, è stata inserita la funzione `return 0`, che svuota lo stack nel caso non fosse stato fatto precedentemente. Tutte le funzioni che utilizzano la ricorsione prevedono la chiamata a `print` per svuotare lo stack.

## 3 Funzione "Add\_Student" e "check\_matricola"

### 3.1 Add\_Student

La funzione Add\_Student permette di poter inserire nome e cognome dello studente e controlla che non vengano utilizzati numeri, ma solo lettere. Lo stesso controllo viene effettuato sulla matricola, che esclude l'utilizzo di lettere, ma solo la presenza di numeri. Tale controllo viene eseguito utilizzando la funzione until `[[ $nome_stud = [$ 0-9] ]]` e until `! [[ = [$ 0-9] ]]`, ovvero considera in input solo i valori non compresi da 0-9 e successivamente tutti i valori che sono solo compresi tra 0-9.

### 3.2 Check\_matricola

La funzione check\_matricola utilizza la funzione grep con i flag `-n -r -q` che cerca per numero di linea in modo ricorsivo e silenziosamente, ovvero senza mostrare l'output sul terminale, che la matricola non sia già stata inserita nel nostro file. Nel caso fosse presente, l'utente viene re-indirizzato a tornare al menu per aggiungere uno studente con una matricola diversa, considerata l'unicità di tale valore, oppure ad uscire dal programma.

## 4 La funzione "Search\_Student"

Questa funzione utilizza la stessa logica del check\_matricola e cerca, invece, la quantità di studenti con il cognome inserito in input dallo user, in cui il flag `-w` del grep permette di cercare per parole intere. Successivamente si passano le righe trovate in un file temporaneo, che viene utilizzato dal ciclo while per leggere e stampare gli utenti che possiedono la variabile ricercata. Infine, tramite il comando `rm` si rimuove il file temporaneo creato.

Nel caso il cognome non dovesse risultare nel file, l'utente può decidere se inserirlo, ritornare al menù, uscire dal programma, oppure cercare un nuovo cognome.

## 5 Funzione "Rm\_Student"

In questa funzione l'utente inserisce in input la variabile della matricola, che viene salvata come una stringa (compare\_var=\$ { # mtdel}). Viene eseguito un controllo sulla lunghezza della variabile inserita e se rientra tra i parametri voluti (`0<stringa<6`). Si utilizza la funzione grep con i flag `-i -h -v` (ovvero, non case-sensitive, ignorando possibili headers dovuti alla stampa dell'elenco ed escludendo il pattern inserito (cioè la variabile)) e si salvano le righe senza la variabile inserita in un file temporaneo, che viene successivamente copiato nel file inserito nello script tramite il comando `cp`. (`cp filerimossi.txt ~ /$filename`)

## 6 Funzione "E\_Passwd" e "shell\_choice"

### 6.1 E\_Passwd

La funzione E\_Passwd permette di poter esportare gli studenti inseriti nel file e formattarli nello stesso modo del file situato in `/etc/passwd`. In input vengono inseriti il path in cui si vuole salvare il file esportato, lo user ID, che sarà un valore crescente, il group ID che riconosce il gruppo degli studenti e il percorso della shell. Si imposta una variabile per contare (`var_tail=1`) che si utilizzerà per tagliare il file del numero di righe esatto che si creeranno dopo che il passwd sarà esportato. L'inizio del ciclo while permette di leggere il filename riga per riga e di sfruttare la variabile "aumenta" per poter stampare tali righe e tagliare il file a livello della prima, seconda, terza o quarta colonna, a seconda della variabile che si intende inserire in output (i.e. `var_nome` taglia a livello della colonna 1 dal delimitatore ";"). Si inseriscono in output le variabili come richiesto dalla consegna e, tramite l'utilizzo di `tail -$ { var_tail } ~/$ dirchoice > my_temp_passwd.txt` si re-indirizza l'output su un file temporaneo che viene infine copiato nel file indicato dal pathway inserito dall'utente ed eliminato tramite il comando `rm`.

### 6.2 Shell\_choice

La funzione shell\_choice legge l'input del percorso della shell e controlla che esista nel path `/etc/passwd` e `/etc/shells` tramite la funzione grep, che impostato i flag `-n -q -r -i -w` (cerca in modo ricorsivo, riga per riga, non case sensitive, "silenzioso", ovvero senza output e controlla parola per parola), cioè verifica se esistono delle shell attive. Nel caso non dovessero esistere, suggerisce all'utente 10 shell indentificate dal percorso `/etc/passwd`, tagliando con un delimitatore "/" a livello del secondo delimitatore fino al decimo delimitatore successivo; stampa solo le ultime dieci righe con il comando `tail -10`.