

Università degli Studi di Napoli Federico II

Corso di Laurea in Informatica

Laboratorio di Programmazione (Gr.3)

Compito del 20/07/2023

Dott. Andrea Apicella

REGOLE

Per questo esame non potete usare il vostro solito account. Per accedere al pc dovete:

1. verificate il nome della macchina, che è la voce in alto a destra, ad esempio *ilc3-44*
2. usate il nome della macchina come nome utente, come password usate *infolab*

Una volta entrati, create una cartella con il vostro nome, cognome e numero di matricola all'interno della cartella *utente (Z:)* che trovate sul desktop. La cartella *utente* monta sul pc locale la vostra home sul server remoto linux. Durante la prova, salvate tutti i vostri file all'interno di questa cartella. **Non salvate direttamente sul desktop perché se si spegne la macchina perdetevi tutto.**

Prima di iniziare materialmente lo svolgimento della traccia, è fortemente consigliato verificare che i file sorgente vengano salvati correttamente dall'editor che scegliete di usare.

Il compito andrà svolto utilizzando preferibilmente il sistema operativo Linux, un editor di testo generico di vostra scelta ed il compilatore *gcc* da linea di comando aperto in una shell.

Una volta che avete finito, la consegna avviene creando un archivio zip della cartella avente come nome *nome_cognome_numero di matricola.zip* e copiandolo nella cartella *consegna elaborati*. State attenti che **questa operazione può essere fatta solo una volta.**

Non potete né leggere, né sovrascrivere, né rinominare nulla di ciò che c'è nella cartella *consegna elaborati*, quindi non preoccupatevi se non potete accedere al compito una volta che avete consegnato.

TRACCIA DEL 20/07/2023, TEMPO: 2.5 ore

Si ricorda che un path assoluto di un file, in linux, è una stringa nel seguente formato:

`/dir1/dir2/.../dirn/nomefile.est`

esempi di path validi:

`/documenti/fileprivati/tracciatrafugata.txt`
`/miafoto.png`
`/db/vegeta.jpg`

Un path assoluto di un file è quindi ben formato se:

- contiene almeno un carattere '/' e tale carattere è il primo;
- l'ultimo carattere è una lettera.

Il nascente social network *Pezzottangram* contiene le *Foto* degli utenti in specifiche strutture dati così formate:

- *img.path*: path assoluto dell'immagine sul supporto di memorizzazione;
- *didascalia*: messaggio di testo di al più 31 caratteri; tale campo non contiene caratteri di a capo al suo interno;
- *n.like*: numero di utenti che apprezzano la foto;

Un file è un'immagine se il suo path assoluto termina con `'bmp'`, `'jpg'`, oppure `'png'`. Per un singolo utente, le informazioni sono memorizzate in un file di testo nel seguente formato:

```
path1
didascalia1
n_like1
path2
didascalia2
n_like2
```

```
.
```


Esempio:

```
/data/miefoto/fototramonto.png
che bel tramonto!
4
/data/lenottibianche.jpg
ho finalmente finito di leggere questo capolavoro della letteratura russa
1
/data/miefoto/paninomangiateapranzo.bmp
gnam!
1546
```

Punto 1: implementare una struttura dati *Bacheca* con politica di accesso *Pila* (*Last In First Out*) in grado di memorizzare delle *Foto*. Tale struttura dovrà fornire almeno le seguenti funzioni:

- *push(Bacheca, Foto)*: inserisce la foto passata come parametro all'interno della bacheca, rispettando la politica di accesso;
- *pop(Bacheca)*: restituisce il collegamento (e rimuove dalla bacheca) la prossima foto;
- *len(Bacheca)*: restituisce il numero di elementi contenuti in bacheca;
- *is_empty(Bacheca)*: restituisce 1 se la bacheca è vuota, 0 altrimenti.

E' a scelta dello studente decidere se tale struttura dati dovrà essere implementata attraverso un array di puntatori a *Post* allocato dinamicamente o attraverso un array di *Post* allocato dinamicamente.

Punto 2: scrivere una funzione *is_valid_path(path)* che, dato un path assoluto in input, restituisca 1 se il path è ben formato, 0 altrimenti

Punto 3: scrivere una funzione *load_user_from_file(Bacheca, nomefile)* che carica tutti i dati contenuti nel file di testo di nome *nomefile* all'interno della *Bacheca*. Il file di testo è fornito con la traccia, ed ha nome "utente42.txt". L'accesso alla bacheca deve essere effettuato sfruttando soltanto le funzioni definite al punto 1.

Punto 4: nella funzione *main()*, caricare nella *Bacheca* solo le foto che hanno path assoluto valido, quindi stampare tutti gli elementi della bacheca uno per volta, dando la possibilità all'utente di scegliere se:

- fermarsi;
- andare alla prossima Foto;
- aggiungere un like alla foto attuale.

L'accesso alla bacheca deve essere effettuato sfruttando soltanto le funzioni definite al punto 1. I post dovranno essere nuovamente presenti all'interno della struttura dati *Bacheca* al termine dello scorrimento (NB: ricordate che è una pila...). E' consigliato implementare una funzione di supporto *print(Foto)* che stampi il contenuto di una singola Foto passata come argomento.

Punto 5: compilare ed eseguire il programma da linea di comando tramite *gcc*. Riportare su di un file di testo di nome "istruzioni.txt" i comandi necessari per effettuare la generazione del file eseguibile. Le compilazioni dei file contenenti le funzioni diverse dal *main()* dovranno essere effettuate *distintamente*. La compilazione del file contenente il *main()* dovrà essere effettuata assieme al linking con tutti i rimanenti file oggetto. Il file eseguibile dovrà avere nome *utente_numeroidmatricola.eseguibile*. Consegnare soltanto i file sorgenti, escludendo i file oggetto ed eseguibili dalla consegna.

Punto Facoltativo:: scrivere una funzione *stats_on_file(nomefile, Bacheca)* che stampi su un file di testo "statistica.txt" un diagramma a barre fatto di asterischi dei like ricevuti da ogni foto. Esempio: se ci sono 3 foto che hanno rispettivamente 5, 4, e 7 like, il diagramma a barre sarà

```
5 *****
4 *****
7 *****
```