

labso2022-2--esame--info+testo
ISTRUZIONI PRATICHE
Esame del modulo di laboratorio di "Sistemi Operativi"

Durata: 120' (2 ore)

- Creare una cartella principale denominata con il proprio numero di matricola e dentro tutti i contenuti richiesti dal compito oltre a rispondere alle domande previste nel modulo online.
- Questa cartella andrà consegnata "zippandola" (compressione formato "zip") in modo da creare un file avente per nome il proprio numero di matricola più l'estensione ".zip". Deve essere compressa l'intera cartella e non solo il suo contenuto.

Se il proprio numero di matricola fosse 123456 questo deve essere anche il nome della cartella e l'archivio compresso da consegnare deve chiamarsi 123456.zip.

- Consegna:
 - Dopo 50' ed entro 60' dall'inizio della prova si deve fare una prima consegna (lavoro parziale) con il lavoro compiuto complessivo fino a tale momento ANCHE SE NON FUNZIONANTE utilizzando il modulo online che sarà indicato.
 - Dopo 90' ed entro 120' dall'inizio della prova si deve fare una seconda consegna (lavoro finale) utilizzando il modulo online che sarà indicato (diverso dal precedente): questa consegna è l'unica considerata per la valutazione finale. Rispondere inoltre alle domande valide per la valutazione.
 - I moduli sono "moduli Google": effettuare il login con il proprio account universitario e inserire tutti i dati richiesti (orientativamente: nome, cognome, numero di matricola e file "zip" da allegare)

NOTA: parte delle verifiche può avvenire con procedure completamente o parzialmente automatizzate per cui le denominazioni e gli output devono essere rigorosamente aderenti alle indicazioni. Dove sono indicate stringhe con sequenze di "escape" vanno rispettate, ad esempio generare un output tipo "7\n" significa che sono presenti esattamente 2 bytes: quello che rappresenta il carattere "7" e un "a capo" (sequenza "\n". Il carattere "a capo" può essere rappresentato anche dal simbolo ↵). Inoltre, non devono essere stampati caratteri aggiuntivi oltre a quelli specificati nel testo. Se per esempio viene chiesto di stampare a video un ID, l'output deve essere "<id>" e non "ID: <id>".

Eventuali irregolarità comportano l'esclusione dalla prova oltre a possibili sanzioni disciplinari.

labso2022-2--esame--info+testo

All'interno della cartella di lavoro realizzare delle applicazioni scrivendo i file sorgente (con estensione ".c") di seguito indicati. Tutti gli eseguibili lavorano sulle cosiddette "named pipe", di seguito indicate semplicemente come "fifo".

VINCOLI:

- la cartella finale deve contenere al più solo il "Makefile" (**così denominato**), i file sorgenti indicati di seguito più un eventuale file "lib.c" opzionale se si ritiene utile disporne.

ATTIVITÀ:

- [4 punti] realizzare il Makefile per la compilazione di tutti i file sorgente in degli eseguibili con lo stesso nome senza l'estensione ".c" (es. un sorgente "main.c" deve compilirsi in un eseguibile denominato "main"). Digitando "make <eseguibile>" si deve generare l'eseguibile denominato <eseguibile> (ad esempio make fifoget dovrebbe compilare il sorgente fifoget.c generando l'eseguibile fifoget).
- Realizzare gli eseguibili di seguito indicati, che accettano esattamente due argomenti: il primo è un percorso, indicato come <path>, che rappresenta un riferimento a una "fifo". Il secondo argomento è un intero, indicato con <n>, nell'intervallo 0-10 (estremi compresi). Tutti gli eseguibili cercano di accedere alla "fifo", terminando con codice di uscita 2 se non è possibile per qualunque motivo. Si può assumere che nessun altro processo acceda alla stessa "fifo" in lettura in concorrenza, che al massimo siano scritti 10 dati dentro la "fifo" e che i dati siano caratteri singoli rappresentanti una lettera maiuscola (A-Z) o una cifra (0-9). Recuperare dati "fino a che è possibile" significa accedere in lettura fino a una condizione di impossibilità (errore, eliminazione o chiusura della fifo, ...)

- [4 punti] fifoget <path> <n>

recupera fino a <n> dati dalla fifo <path> e li manda su stdout separati da un "\n", terminando con codice di uscita 0 se li trova tutti o con codice di uscita 1 se si termina prima.

- [5 punti] fifoskp <path> <n>

recupera dati fino a che sia possibile mandandoli su stdout separati da un "\n", saltando però il dato <n>-esimo e terminando con codice di uscita 0 se ne trova almeno <n> (quindi se ha stampato almeno <n>-1 dati) o con codice di uscita 1 altrimenti.

- [5 punti] fiforev <path> <n>

recupera fino a <n> dati dalla fifo <path> e li manda su stdout in ordine inverso separati da un "\n", terminando con codice di uscita 0 se li trova tutti o con codice di uscita 1 se si ha un errore prima (nel qual caso manda comunque in stdout in ordine inverso quelli recuperati)

- [7 punti] fifoply <path> <n>

recupera dati fino a che è possibile o fino a incontrare <n> cifre, mostrando i dati alfabetici incontrati in ordine inverso su stdout separati da "\n" e poi una stringa del tipo " $n_1+n_2+\dots+n_j=t$ " dove i vari n_i sono i dati rappresentanti le cifre recuperate, ma in ordine inverso, e t è il totale (somma aritmetica). Il codice di uscita deve essere 0 se si è terminato perché si sono incontrate <n> cifre (quindi $j=n$), altrimenti deve essere 1 (in questo caso si devono comunque mandare su stdout tutti i dati recuperati fino a quel momento con lo stesso criterio).

Esempio:

"./fifoply <fifo> 4" con input su pipe "ab3cd6ef74" restituisce il seguente output:

```
f
e
d
c
b
a
4+7+6+3=20
```