

Prova Pratica 019

turno 1 gruppo 1

2017 febbraio 20

i file da consegnare **devono** essere collocati nella directory **CONSEGNA** dentro la home directory dell'utente studente.

Prova Pratica 019 - turno 1 gruppo 1

Download Materiale:

Scaricare il file con le **dispense** e gli **esempi** svolti a lezione

```
wget http://esameso.csr.unibo.it/TREE4OS1617.tgz
```

Decomprimere l'archivio scaricato: `tar xvzf TREE4OS1617.tgz`

Viene creata una directory **TREE4OS1617** con dentro una sottodirectory **sistemioperativi** con dentro tutto il **materiale**.

Potete navigare tra il materiale con un normale browser aprendo l' URL

file:///home/studente/TREE4OS1617/sistemioperativi/dispenseSistOp1617.html

Esercizi d'esame: per chi ha difficoltà a superare la prova pratica, ho previsto due tipi di prove:

- A. una prova **COMPLICATA**, e' la modalità normale che vi permette di raggiungere un **voto massimo** (nella prova pratica stessa) di **30Lode** ,
- B. ed una prova **SEMPLICE**, un po' **meno complicata**, che però vi permette di raggiungere un **voto massimo di 24** perché l'esercizio di programmazione concorrente é meno difficile.

Scegliete voi quale prova svolgere in funzione della vostra preparazione.

La prova **COMPLICATA** è composta dagli esercizi **52 e 53**,

La prova **SEMPLICE** è composta dagli esercizi **51 e 53**.

Come vedere l'esercizio 53 è comune alle due prove.

Svolgete **SOLO** gli esercizi della prova che vi interessa.

I file da consegnare **devono** essere collocati nella directory **CONSEGNA** dentro la home directory dell'utente studente.

Esercizio Esame Pratica - 51 - avvisa (semplice)

Un programma crea inizialmente **10** thread **avvisa**, ciascuno dei quali creerà un thread figlio di tipo **avvisa**, il quale, a sua volta creerà un thread figlio **avvisa** e così via all'infinito.

Il main deve continuamente rimanere in attesa di ricevere da ciascuno dei thread **avvisa** in esecuzione il thread identifier di quel thread e poi deve chiamare la join per quel thread.

Più nel dettaglio, ciascun thread di tipo **avvisa** deve essere joinable e :

- appena inizia deve aspettare 1 secondo,
- poi deve passare al main il proprio thread identifier,
- poi deve creare un figlio,
- poi deve terminare.

NB: attenti a non perdersi dei thread identifier!

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX ed avvalendosi delle opportune strutture dati per la sincronizzazione. Scrivere il Makefile per compilare e linkare i sorgenti. La mancanza del Makefile viene considerato un errore grave.

Occorre inserire il controllo di errore nelle chiamate a funzione delle librerie dei pthread. In caso di errore grave, terminare il programma producendo un avviso a video.

Esercizio Esame Pratica - 52 - conigli (complicato)

Un programma crea inizialmente **5** thread di tipo **coniglio**.

Ciascun thread coniglio si accoppia con un altro thread coniglio.

Dopo l'accoppiamento ciascuno dei thread coniglio crea un altro thread coniglio e poi termina.

L'accoppiamento avviene secondo queste regole:

- Per l'accoppiamento viene usata una sola ed unica tana, che all'inizio è vuota.
- Nella tana possono entrare al massimo due conigli.
- Per l'accoppiamento occorrono due conigli insieme nella tana.
- Quando due conigli sono nella tana l'accoppiamento è istantaneo.
- Prima che due altri conigli possano entrare, entrambi i conigli che erano precedentemente nella tana devono essere usciti.

Modellare ed implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (coniglio) ed avvalendosi delle opportune strutture dati per la sincronizzazione. Scrivere il Makefile per compilare e linkare i sorgenti. La mancanza del Makefile viene considerato un errore grave.

Occorre inserire il controllo di errore nelle chiamate a funzione delle librerie dei pthread. In caso di errore grave, terminare il programma producendo un avviso a video.

PRECISAZIONE sul prossimo Esercizio **53 - cercaemail**

Nel prossimo esercizio 53, vi si chiede di usare due file `matricola.txt` ed `email.txt` forniti nel file `archivio019.tgz`

I diversi campi nelle righe dei due file sono separati da **TABULAZIONI (TAB)**.

Se vi viene più facile, potete modificare quei file mettendo degli spazi (**SPACE**) al posto del **TAB**.

Fate come credete sia più facile per voi

Esercizio Esame Pratica - 53 - cercaemail

Sia dato un file di testo `matricola.txt` in cui, in ciascuna riga e' contenuto il nome di uno studente, il suo cognome, il suo numero di matricola e il suo corso di laurea. I vari campi sono separati da tabulazioni (tab). Non esistono due studenti con la stessa matricola. Possono esistere piu' studenti con lo stesso nome, ma devono per forza avere matricole diverse. Vedere esempio a lato ----->

Sia dato un file di testo `email.txt` in cui, in ciascuna riga, e' contenuto il numero di matricola di uno studente e la sua email universitaria, separati da tab. Non possono esistere due email uguali. Non possono esistere due matricole uguali. Vedi esempio a lato ---->

(esempi contenuti in archivio)

esempio file	matricola.txt
Luca Andreucci	578 informatica
Vittorio Ghini	666 teologia
Giovanni Pau	999 informatica
Luca Andreucci	123 lettere

esempio file	email.txt
578	LA@studio.unibo.it
666	VG@studio.unibo.it
999	GP@studio.unibo.it
123	LA2@studio.unibo.it

Scrivere uno script **cercaemail.sh** a cui vengono passati 3 argomenti, rispettivamente il NOME, il COGNOME e la MATRICOLA di uno studente. Lo script usa al suo interno il file `email.txt`, scopre la EMAIL dello studente i cui dati sono stati passati come argomenti, e stampa a video la stringa:

la email dello studente NOME COGNOME MATRICOLA e' EMAIL

dove NOME COGNOME MATRICOLA e EMAIL sono rispettivamente il nome, il cognome, la matricola e la email dello studente.

Scrivere poi uno script **tutteemail.sh** che utilizza al suo interno il file `matricola.txt`. Lo script legge una dopo l'altra tutte le righe del file, estrae le informazioni NOME COGNOME MATRICOLA di ciascuno studente, e per ciascuno studente chiama lo script `cercaemail.sh` passandogli come argomenti il NOME COGNOME MATRICOLA dello studente.