

Sistemi Operativi CANALE M-Z
Compito Scritto del 03 Dicembre 2018

Cognome=_____, Nome=_____
Matricola=_____

Durata 3 ore

Domanda 1: max 5 punti

Illustrare le architetture tipiche di un SO. **(punti 2)**

Illustrare l'architettura ibrida di Linux dettagliando la gestione dei moduli kernel. **(punti 3).**

Si scriva la risposta in un file di testo denominato DOMANDA1_COGNOME_NOME

Domanda 2: max 5 punti

Illustrare in maniera dettagliata i segnali, la loro gestione da parte del kernel e i loro limiti. **(punti 2)**

Scrivere un programma in C completo che dimostri i limiti precedentemente evidenziati. **(punti 3).**

Si scriva la risposta in un file di testo denominato DOMANDA2_COGNOME_NOME

Domanda 3: max 20 punti

Si vogliono realizzare due processi NON IMPARENTATI (ossia INDIPENDENTI) che producono e consumano informazioni condivise tramite il meccanismo della *Memoria Condivisa*. In particolare i due processi condividono un'area di memoria che **contiene uno struct costituito da: un vettore di 10 float e una stringa di 10 caratteri**. Inoltre entrambi i due processi utilizzano uno stesso file di nome "output".

I due processi sono sincronizzati, nel senso che le azioni che compiono seguono un ordine ben preciso: prima opera il primo processo e quando finisce opera il secondo processo. I due processi lavorano secondo un ciclo infinito che può essere interrotto solo dal primo processo. In particolare il primo processo, chiede all'utente se vuole continuare; nel caso in cui l'utente non voglia continuare deve inserire la parola "**fine**", che verrà scritta nel campo stringa dello struct condiviso tra i due processi. L'inserimento di questa stringa comporta la fine del primo processo e, visto che lo struct è condiviso tra i due processi, anche il secondo processo termina quando legge il valore "fine" all'interno dello struct condiviso.

Il primo processo inizia per prima; la prima cosa che fa è quello di aprire il file "output" e se esso contiene elementi (alla prima esecuzione non conterrà nulla ovviamente), devono essere tutti letti (a partire dall'inizio del file fino alla fine del file) e stampati a video. Poi il primo processo riempie in modo casuale il vettore di 10 elementi contenuto nello struct condiviso, e passa il turno al secondo processo.

Il secondo processo legge il campo stringa dello struct condiviso e se contiene il valore "fine", termina. Altrimenti, il secondo processo legge tutto il vettore dello struct condiviso e scrive tutti gli elementi nel file di nome "output". Si suppone che il secondo processo apra il file con l'opzione O_TRUNC in modo da svuotarne il contenuto e riempirlo ex novo. Il file viene così riempito di tutti i 10 elementi contenuti nel vettore condiviso.

A questo punto il turno ripassa al primo processo che ripete quanto detto prima per poi passare il turno al secondo processo, come detto, e così via fino a quando i due processi terminano perché l'utente del primo processo ha inserito la parola "fine", come già detto.

Per lo svolgimento dell'esercizio si utilizzino la **memoria condivisa, i semafori SystemV e i file**.

Il nome dei programmi prodotti devono contenere il nome e cognome dello studente.