Report Sistemi Operativi

DisastrOS Semaphores

PEPE SVEVA
1743997
pepe.1743997@studenti.uniroma1.it

1 Intro

Implementazione dei semafori in DisastrOS. Sono state implementate 4 funzioni:

1. `int disastrOS_semopen (int id, int count)`
2. `int disastrOS_semclose (int fd)`
3. `int disastrOS_semwait (int fd)`
4. `int disastrOS_sempost (int fd)`

2 disastrOS_semopen

La `int disastrOS_semopen (int id, int count)` prende come parametri **id e count**, i quali sono interi e rappresentano id del semaforo e il valore che esso assume. Il semaforo diventa cosi accessibile tramite il suo id. Il count non può essere negativo al momento dell' apertura del semaforo.

La funzione crea un nuovo semaforo o ne apre uno già esistente con id uguale al parametro della syscall. Poi vado ad aggiungere un descrittore `SemDescriptor sem_desc` e `SemDescriptorPtr sem_desc_ptr` alla lista dei semafori aperti dal processo che è in esecuzione.

In caso di successo la funzione ritornerà un numero intero ≥ 0, altrimenti ritornerà un errore (intero negativo).

3 disastrOS_semclose

la `int disastrOS_semclose (int fd)` prende come parametro **fd**, il quale corrisponde al file descriptor del semaforo, infatti nella semopen setto come valore di ritorno della syscall il file descritptor.

La funzione chiude il semaforo e libera il `SemDescriptor sem_desc` e `SemDescriptorPtr sem_desc_ptr` associati a quel semaforo; cioè vado a togliere SemDescriptor dalla lista dei SemDescriptor del processo e SemDescriptorPtr dalla lista dei SemDescriptorPtr del semaforo.

In caso di successo la funzione ritornerà un numero intero ≥ 0, altrimenti ritornerà un errore (intero negativo).

4 disastrOS_semwait

la `int disastrOS_semwait (int fd)` prende come parametro **fd**, il quale corrisponde al file descriptor del semaforo, infatti nella semopen setto come valore di ritorno della syscall il file descritptor.

La funzione va a decrementare il valore count (contatore) del semaforo. Se risulta che il valore dei count < 0, il processo viene nella `waiting_list` ed inoltre viene inserito il `SemDescriptorPtr sem_desc_ptr` nella lista `waiting_descriptors` del semaforo.

In caso di successo la funzione ritornerà un numero intero ≥ 0, altrimenti ritornerà un errore (intero negativo).

5 disastrOS_sempost

la `int disastrOS_semwpost (int fd)` prende come parametro **fd**, il quale corrisponde al file descriptor del semaforo, infatti nella semopen setto come valore di ritorno della syscall il file descritptor.

La funzione va a incrementare il valore count (contatore) del semaforo. Se risulta che il valore dei count *le* 0, viene tolto il `SemDescriptorPtr sem_desc_ptr` nella lista `waiting_descriptors` del semaforo e il processo corrispondente verrà messo nella `ready_list`.

In caso di successo la funzione ritornerà un numero intero ≥ 0, altrimenti ritornerà un errore (intero negativo).

6 Gestione Errori

`DSOS_COUNT_SEM_NEGATIVE` $\rightarrow$ il count del semaforo è negativo, non è possibile avere count negativo quando si deve aprire un semaforo.

`DSOS_SEM_NOT_ALLOC` $\rightarrow$ errore nell'allocazione del semaforo.

`DSOS_SEM_DESC_NOT_ALLOC` → errore nell'allocazione del descrittore (SemDecriptor) del semaforo.

`DSOS_SEM_DESC_PTR_NOT_ALLOC` → errore nell'allocazione SemDescriptorPtr del semaforo.

`DSOS_SEM_DESC_NOT_FOUND` → file descriptor passato come parametro non è relativo a nessun il descrittore del semaforo, cioè a quel file descriptor non è associato a nessun semaforo aperto dal processo.

`DSOS_SEM_NOT_FOUND` → file descriptor passato come parametro non è associato a nessun semaforo aperto dal processo.

`DSOS_SEM_DESC_PTR_NOT_FOUND` → file descriptor passato come parametro non è relativo a nessun il SemDescriptorPtr del semaforo, cioè a quel file descriptor non è associato a nessun semaforo aperto dal processo.

7 Esecuzione programma

Compilare programma: `make`

Eseguire programma: `./disastrOS_test`