

\*\*\*\*\*Semaphores-disastrOS\*\*\*\*\*

- WHAT:

Il progetto riguarda l'accesso asincrono dei processi alle risorse disponibili, cioè avviene e mediante l'implementazione del modello produttore - consumatore attraverso l'uso di semafori e delle funzioni che ne permettono la corretta esecuzione. Il risultato finale del test consiste nella scrittura ordinata e nella lettura di un buffer circolare avente 50 elementi.

- HOW:

Le quattro systemcall sotto descritte gestiscono il funzionamento dei semafori:

1. semOpen: necessaria alla creazione di un semaforo identificato da un id e un contatore, denominati rispettivamente 'sem\_id' e 'sem\_count'.  
Nel caso in cui il semaforo fosse già presente nella lista dei semafori allora avviene la semplice apertura di un nuovo descrittore altrimenti viene effettuata l'allocazione e l'aggiunta del semaforo alla lista dei semafori.  
Viene inoltre creato un puntatore al file descrittore e viene incrementato il contatore dei descrittori attivi del processo in esecuzione.  
Infine se non ci sono stati errori viene impostato come valore di ritorno il file descriptor del semaforo appena aperto (altrimenti il valore di ritorno dipende dal tipo di errore stampato).

2. semClose: necessaria alla rimozione e alla deallocazione, dalla lista dei descrittori del processo in esecuzione, del file descrittore passato come argomento in input.  
Viene inoltre rimosso e deallocato il puntatore al descrittore nella lista dei descrittori attivi del semaforo e, se il semaforo non ha più descrittori attivi e/o in attesa, allora viene rimosso e deallocato dalla lista globale.  
Infine viene decrementato il contatore dei descrittori attivi e, se non ci sono stati errori, viene impostato come valore di ritorno 0 (altrimenti il valore di ritorno dipende dal tipo di errore stampato).

3. semWait: necessaria a decrementare il contatore del semaforo associato al descrittore passato. Quando il contatore assume valori negativi, il puntatore al descrittore e il PCB vengono spostati dalla ready alla waiting list e un nuovo processo viene messo in esecuzione. Infine se non ci sono stati errori, viene impostato come valore di ritorno 0 (altrimenti il valore di ritorno dipende dal tipo di errore stampato).

4. semPost: necessaria a incrementare il contatore del semaforo associato al descrittore passato. Se il contatore, a seguito dell'incremento, possiede valore non positivo, il puntatore al descrittore e il PCB vengono spostati dalla waiting alla ready list e lo stato di un nuovo processo viene impostato a Ready. Infine se non ci sono stati errori, viene impostato come valore di ritorno 0 (altrimenti il valore di ritorno dipende dal tipo di errore stampato).

- HOW TO RUN:

Per eseguire il test è necessario compilare tutti i files mediante il comando da terminale 'make' e quindi eseguire il file di prova con './disastrOS\_test'.