

Prova pratica di Calcolatori Elettronici (nucleo v6.*)

C.d.L. in Ingegneria Informatica, Ordinamento DM 270

23 luglio 2015

1. Prevediamo che un processo possa creare delle zone di memoria, dette **shmem**, ciascuna con un identificatore unico. I processi che vogliono accedere ad una **shmem** devono aggiungerla al proprio spazio di indirizzamento, specificandone l'identificatore. Una volta aggiunta, la **shmem** sarà disponibile contigualmente all'interno della parte utente/condivisa dello spazio di indirizzamento del processo. Un processo può aggiungere più **shmem** al proprio spazio e le diverse **shmem** non devono sovrapporsi. Non è importante che i processi che condividono una stessa **shmem** la vedano tutti allo stesso indirizzo. In qualunque momento, un processo può eliminare dal proprio spazio di indirizzamento una **shmem** precedentemente aggiunta.

Per descrivere una **shmem** aggiungiamo al nucleo la seguente struttura dati:

```
struct des_shmem {
    natl npag;
    natl first_frame_number;
};
```

Il campo **npag** contiene la dimensione (in pagine) della **shmem**. Tutte i frame che contengono la **shmem**, nell'ordine in cui devono comparire nella memoria di tutti i processi che la condividono, sono mantenuti in una lista; il numero del frame in testa alla lista è contenuto nel campo **first_frame_number**. Ogni frame punta al successivo tramite un nuovo campo **natl next_shmem** che abbiamo aggiunto ai descrittori di frame.

Inoltre, aggiungiamo i seguenti campi ai descrittori di processo:

```
vaddr avail_addr;
des_attached *att;
```

Il campo **avail_addr** contiene il primo indirizzo libero nella parte utente/condivisa del processo. Tutti gli indirizzi da **avail_addr** fino a **fin_utn_c** (escluso) sono disponibili per contenere zone di memoria condivisa. Il campo **att** è la testa di una lista di elementi di tipo **des_attached**, il cui scopo è di tener traccia di tutte le **shmem** a cui il processo è collegato. Il tipo **des_attached** è così definito:

```
struct des_attached {
    natl id;
    vaddr start;
    des_attached *next;
};
```

Il campo **id** è l'identificatore di una **shmem** **shmem**; il campo **start** è l'indirizzo virtuale (nello spazio di indirizzamento del processo) a partire dal quale questa **shmem** è visibile. Il campo **next** punta al prossimo elemento della lista.

Aggiungiamo infine le seguenti primitive:

- `natl shmем_create(natl npag)` (tipo 0x5c, già realizzata): Crea una nuova zona di memoria condivisibile tra più processi, grande `npag` pagine, e ne restituisce l'identificatore.
- `vaddr shmем_attach(natl id)` (tipo 0x5d, già realizzata): Permette ad un processo di aggiungere la `shmем id` al proprio spazio di indirizzamento e ne restituisce l'indirizzo di partenza.
- `void shmем_detach(natl id)` (tipo 0x5e, da realizzare): Permette ad un processo di eliminare la `shmем id` dal proprio spazio di indirizzamento. Abortisce il processo se la `shmем id` non è tra quelle a cui il processo è collegato.

Modificare i file `sistema.cpp` e `sistema.S` in modo da realizzare le primitive appena descritte.

ATTENZIONE: nella `shmем_detach()`, tralasciare la gestione di `avail_addr`.