

# Prova pratica di Calcolatori Elettronici (nucleo v6.\*)

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

24 novembre 2021

1. Le parti utente/convisa e io/condivisa della memoria virtuale dei processi contengono il codice, i dati globali e lo heap del modulo utente e del modulo I/O, rispettivamente. I due heap hanno una dimensione fissa, data dalle costanti `DIM_USR_HEAP` e `DIM_IO_HEAP`. Il primo indirizzo che cade al di fuori dello heap è detto *break* (utente o I/O). Vogliamo estendere il nucleo in modo da fornire al modulo utente e al modulo I/O la possibilità di spostare il proprio break avanti o indietro a piacimento, pur restando nei limiti delle rispettive zone condivise, e senza andare a toccare la parte di memoria che contiene i loro dati globali. Dopo che un processo utente ha spostato il break in avanti, tutti i processi utente devono poter accedere alla nuova zona di memoria senza causare page fault. Viceversa, dopo che un processo utente ha spostato il break indietro, gli accessi di qualunque processo utente a indirizzi della zona utente/condivisa che cadano oltre il break devono causare page fault (e analogamente per il break del modulo I/O, con riferimento agli accessi delle primitive di I/O o dei processi esterni).

Nel nucleo modificato i due heap hanno inizialmente dimensione nulla (il break coincide con il primo indirizzo che non appartiene alle sezioni dati dei due rispettivi moduli). Aggiungiamo però le seguenti due primitive (abortiscono il processo nel caso di errore):

- `vaddr incr_break(natq delta)` (da realizzare): incrementa il break (utente o I/O) di `delta` e azzerla la nuova memoria che entra a far parte dello heap; restituisce il nuovo valore del break; un valore di `delta` pari a zero può essere usato per conoscere il valore corrente del break, senza modificarlo;
- `vaddr decr_break(natq delta)` (da realizzare): decrementa il break (utente o I/O) di `delta`; restituisce il nuovo valore del break.

Le due primitive capiscono quale break deve essere spostato in base al livello (sistema o utente) del loro chiamante (che può essere ottenuto con la funzione `liv_chiamante()`). È sempre un errore se `delta` non è multiplo della dimensione di una pagina (con l'eccezione di `delta` pari a zero, che è ammesso). Per `incr_break` è anche un errore se sommare `delta` al break causa un overflow o porta il break fuori dalla zona condivisa; per `decr_break` è un errore se sottrarre il `delta` causa un underflow o fa scendere il break sotto il limite della sezione dati.

Modificare il file `sistema.cpp` in modo da realizzare le primitive e le parti mancanti.

**Suggerimento:** le funzioni `carica_IO` e `carica_utente` sono state modificate in modo da restituire in `mod_end` l'indirizzo minimo ammesso per il rispettivo break.