

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

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

16 febbraio 2022

1. Definiamo una “memory-area” come una zona della memoria privata di un processo  $P_1$  gestita da un secondo processo  $P_2 \neq P_1$ , detto *monitor* di  $P_1$ . Un processo  $P_2$  può diventare il monitor di  $P_1$  tramite una primitiva `ma_attach()`, che definisce anche la dimensione (in pagine) della memory-area di  $P_1$ . Da quel momento in poi, tutti i page-fault generati da  $P_1$  su indirizzi che cadono all'interno della sua memory-area verranno intercettati da  $P_2$ . Il processo  $P_2$  può sospendersi in attesa che  $P_1$  causi un page fault tramite una primitiva `ma_wait()`, che restituisce informazioni sulla causa del page fault. Il processo  $P_1$  non viene distrutto e resta in attesa che  $P_2$  lo riavvii tramite la primitiva `ma_cont()`, eventualmente dopo aver creato una traduzione che risolva il page fault.

Il processo  $P_2$  può manipolare la memory-area di  $P_1$  in qualunque momento, usando una primitiva `ma_map(src, dst, P, W)`. Il parametro `src` deve essere un indirizzo dello spazio utente/condiviso, il parametro `dst` deve essere un indirizzo della memory-area di  $P_1$ , e i parametri `P` e `W` sono due booleani. Sia `src` che `dst` devono essere allineati alla pagina.

Se `P` è `true` la primitiva `ma_map` serve a creare o modificare una traduzione nella memory-area di  $P_1$ . In particolare, la primitiva deve fare in modo che, per il processo  $P_1$ , la pagina `dst` sia tradotta nel frame che corrisponde a `src`. Inoltre, se `W` è `true` il processo  $P_1$  deve poter usare la traduzione anche in scrittura, altrimenti solo in lettura. Se esisteva già una traduzione per `dst`, questa deve essere aggiornata in modo da rispettare la nuova richiesta (per esempio, per vietare la scrittura se prima era concessa). La primitiva provvede anche ad azzerare i bit `A` e `D` nell'entrata di livello 1 relativa a `dst`.

Se `P` è `false` la primitiva serve ad eliminare la traduzione di `dst`. In questo caso `src` e `W` sono ignorati.

In entrambi i casi la primitiva restituisce il precedente valore del byte di accesso dell'entrata di livello 1 relativa `dst` (0 se la traduzione non esisteva). In particolare, il byte di accesso deve contenere i valori che `P`, `A`, `D` e `RW` avevano prima della loro eventuale modifica. Se non è stato possibile creare la traduzione (perché serviva allocare una tabella e lo spazio era esaurito) la primitiva deve restituire `0xffffffffffffffff`.

Modificare il file `sistema.cpp` per aggiungere la primitiva `ma_map()` appena descritta. La primitiva deve abortire il processo chiamante nei seguenti casi di errore: il chiamante non è un monitor; `dst` non è allineato o non appartiene alla memory-area del processo monitorato; `src` (se utilizzato) non è allineato o non appartiene alla parte utente/condivisa, oppure non è accessibile in scrittura nel caso questa sia richiesta.

**Suggerimento:** può darsi che `map()` e `unmap()` non siano sufficientemente espressive per realizzare `ma_map()`. Può essere utile lavorare direttamente con un `tab_iter`. Fare attenzione al contatore di entrate valide delle tabelle (funzioni `inc_ref()` e `dec_ref()`), che deve restare consistente quando si cambia il valore di qualche bit `P`.