

Scelte progettuali

Stefano De Santis, Cristiano Guidotti, Iacopo Porcedda, Jacopo Rimediotti

Alma Mater Studiorum – Università di Bologna

Scelte progettuali

La fase 2 del progetto BiKaya è stata sviluppata in linguaggio C, con l'obiettivo di realizzare il funzionamento di uno scheduler dei processi funzionante per le architetture uARM e uMPS, che estendesse le funzionalità di quello di fase 1.5.

Rimandiamo la lettura della documentazione di 1.5 per le parti invariate.

La sua realizzazione è stata strutturata in 3 moduli:

- Estensione funzionalità dello scheduler
- Gestione delle system call
- Gestione degli interrupt

Nonostante questa strutturazione i moduli interagiscono tra di loro in base alle necessità.

Gestione scheduling dei processi

Lo scheduling dei processi avviene in sinergia con la gestione delle eccezioni.

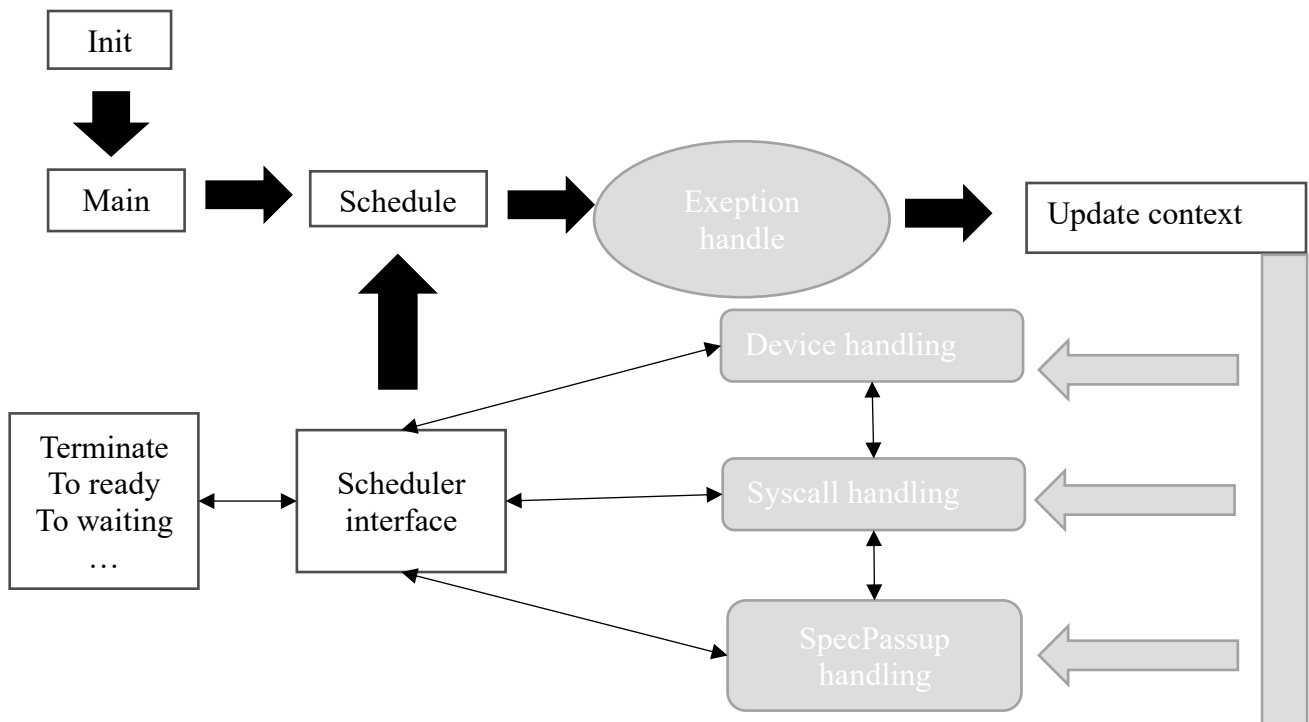
Lo scheduler infatti fornisce delle interfacce agli altri gestori per poter richiamare le funzionalità dello scheduler per proseguire il suo ciclo di vita.

Per poter effettuare varie interazioni con lo scheduler che potrebbero alterare il processo corrente, è necessario che ogni handler richiami un'apposita interfaccia dello scheduler, prima di operazioni specifiche, per aggiornare il contesto del pcb memorizzato sullo scheduler con quello più recente.

Le fasi nuove (o aggiornate) dello scheduler sono:

- Selezione ed esecuzione processo
- Sospensione / Riattivazione processo
- Terminazione processo

Il grafico seguente mostra il ciclo di vita dello scheduler e le transizioni da uno stato ad un altro per mezzo delle sue interfacce.



Selezione ed esecuzione processo (scheduling)

In questa fase lo scheduler si occupa di effettuare un'operazione di aging sui processi nella ready queue, inserisce eventualmente il processo precedente in esecuzione, e successivamente prende il primo PCB disponibile nella ready queue che avrà la priorità maggiore di tutti gli altri, impostando un timer di 3 ms sul processore, entro il quale il controllo dell'esecuzione è assegnato al processo scelto; infine carica lo stato del PCB sul processore, aggiornando il controllo dell'esecuzione.

Nel caso in cui la ready queue sia vuota e non ci sono processi in attesa sui dispositivi, il processore è impostato in stato di HALT.

Se invece almeno un processo è in attesa su qualche coda riservata ad un dispositivo, ma la ready queue è vuota, lo scheduler metterà in esecuzione un processo di idle, fino alla prossima fase di scheduling.

Lo scheduler forzerà lo scheduling di un nuovo processo se è stato notificato esplicitamente, o implicitamente in caso di mancanza del processo corrente; altrimenti lo scheduler invece continuerà a mettere in esecuzione il processo corrente per il suo time slice rimanente. Si noti che questa scelta è fatta solamente nel caso non sia stata effettuata qualche altra operazione che ha rimosso il processo in esecuzione da altri handler. Infatti, nel caso di interruzione per System

call, se non è stata alterata la condizione del processo nello scheduler (ad esempio per syscall non bloccanti), esso può continuare l'esecuzione.

Invece nel caso di interruzione per interrupt, il context switch può essere effettuato se è stato coinvolto il processo corrente, quindi nel caso sia terminato il suo timeslice.

In caso lo scheduler continui ad eseguire lo stesso processo corrente, il tempo dell'elaborazione di qualsiasi handler, non è aggiunto al time slice "perso" dal processo.

Sospensione / Riattivazione processo

In questa fase, in base alla situazione specifica, il processo in esecuzione può essere messo in stato di attesa su qualche coda di un semaforo, oppure inserito nella ready queue se prima si trovava in uno stato differente.

Dopo questa fase lo scheduler implicitamente sceglierà ed effettuerà lo scheduling di un nuovo processo.

Si vuole inoltre menzionare che la priorità del processo corrente è ripristinata a quella originale sia quando viene sospeso, in attesa di qualche semaforo, che quando è re-inserito nella ready queue al termine del suo time slice.

Terminazione processo

In questa fase il processo deve essere terminato con eventualmente tutta la sua progenie, a causa di conseguenze di altri handler.

Lo scheduler rimuove e dealloca il PCB che era in esecuzione e tutta la sua progenie dalla ready queue, o alternativamente, se specificato durante la chiamata della procedura, solo il PCB rendendo orfani tutti i suoi processi figli nella ready queue.

Dopo questa fase lo scheduler implicitamente sceglierà ed effettuerà lo scheduling di un nuovo processo.

Gestione delle system call

Il meccanismo di gestione delle System Call è implementato nei moduli handler del kernel in esame. Quando un processo richiama l'attenzione del sistema operativo attraverso una system call, il gestore tenta di gestire e soddisfare la richiesta, però prima di ogni operazione aggiorna lo stato del pcb in memoria, sia per avere memorizzata la versione più recente, che per tracciare quando il processo entra nel kernel.

Per soddisfare la richiesta, viene selezionata la procedura associata al numero di system call passato come primo parametro.

In caso la system call dovesse restituire un qualche valore, verrà prontamente inserito nel registro di ritorno del processo, in base all'architettura.

Le system call attualmente implementate sono 8; infatti nel caso venga richiesta una system call con codice identificativo superiore, il controllo di esecuzione verrà passato ad un gestore di livello superiore, definibile tramite la system call 7 (SpecPassup), a cui sarà correttamente inizializzata la old area, con lo stato del processo che ha effettuato la richiesta. Se tale gestore di livello superiore non è stato definito sul processo corrente, allora verrà terminato come nella gestione delle altre eccezioni.

Gestione degli interrupt

Il gestore degli interrupt è implementato nei moduli handler relativi alla rispettiva architettura, che richiameranno un gestore condiviso (quasi) "indipendente" dai dettagli dell'architettura.

Tale gestore si occuperà della gestione relativa agli interrupt causati da dispositivi di IO e dal dispositivo di Interval Timer.

Come per gli altri handler, prima della gestione vera e propria, aggiorna il pcb del processo corrente nello scheduler, e prosegue con la gestione dell'interrupt.

In linea di massima, al sollevamento di un interrupt, il gestore identificherà la linea che lo ha generato e, nel caso di un interrupt sollevato da un dispositivo di IO, ne identificherà anche il device, in modo tale da avviarne correttamente la gestione.

Nel caso di interrupt multipli, l'handler si occuperà di gestirli in base alla loro priorità (priorità più alta 0, meno alta 7). Nel caso di interrupt sollevati da terminali, gestirà opportunamente entrambi i subdevice. In caso di una richiesta non valida verrà mandato in PANIC il sistema.

Vale la pena spendere due righe sulla gestione di interrupt relativi a device di IO.

Il gestore, una volta identificata la sorgente dell'interrupt, in questo caso un dispositivo di IO, si occuperà di svegliare l'eventuale processo in attesa di risposta e consegnerà ad esso il risultato dell'operazione di IO richiesta. Al termine della gestione, si occuperà anche di avviare una nuova operazione di IO nel caso in cui ci sia un altro processo in coda in attesa sul semaforo del device; il comando in particolare è recuperato dall'apposito registro di parametro passato tramite system call 6. Chiaramente, data la natura differente dei registri dei terminali e dei registri degli altri dispositivi di IO, la gestione sarà leggermente differente.

Gestione altre eccezioni

Le eccezioni trap, breakpoint, e TLB sono gestite parzialmente in specifici gestori, che in particolare si limitano a inoltrare la richiesta, caricando sul processore lo stato o new area di un gestore di livello superiore, e memorizzando lo stato del processo che ha causato l'eccezione nella sua relativa old area, definite in precedenza sul processo corrente dalla system call 7.

Se sul processo non è stato definito il gestore relativo alla eccezione generata, il singolo processo verrà terminato (quindi non tutta la progenie, facendo diventare orfani gli eventuali processi figli), e verrà schedulato un nuovo processo.

Variazioni rispetto a specifiche

Rispetto alle specifiche richieste, si vuole far notare che le system call 4 e 5 sono state fornite anche di un ulteriore valore di ritorno per informare il chiamante, se le operazioni di Verhogen o Passeren sono state eseguite con successo, oppure no se il semaforo fornito non è valido (NULL).