

PandOS+

Fase 2

Claudio Jin <claudio.jin@studio.unibo.it>

Leonardo Signorini <leonardo.signorini@studio.unibo.it>

Andrea Venturoli <andrea.venturoli5@studio.unibo.it>

Anno accademico

2023 – 2024

Corso di Sistemi Operativi

Alma Mater Studiorum · Università di Bologna

1 Introduzione

La seconda fase del progetto PandOS+ consiste nell'implementazione del livello 3 dell'architettura astratta di un sistema operativo proposta da Dijkstra. In particolare è necessario implementare le funzionalità del kernel per l'inizializzazione del sistema, la schedulazione dei processi e la gestione delle eccezioni.

1.1 Organizzazione dei file

I file coinvolti per l'implementazione delle specifiche sono i seguenti:

pcb	È stata aggiunta una funzione per ciclare sulle liste.
msg	Aggiunto il supporto per nuovi tipi di messaggi.
initial	Gestisce la fase di inizializzazione del sistema e contiene l'implementazione di memcpy.
scheduler	Implementa lo scheduler.
exceptions	Reindirizza le eccezioni ai relativi gestori. Implementa il gestore delle system call.
interrupt	Implementa il gestore degli interrupt.
ssi	Implementa il system service interface per gestire le richieste di servizi da parte dei processi

2 Gestione dei messaggi

La struttura dati dei messaggi è stata modificata per facilitare la trasmissione dei tipi di messaggi più frequenti, ovvero messaggi all'ssi e stringhe per la stampa sui terminali. Questi due tipi vengono gestiti di conseguenza nelle system call SendMessage E ReceiveMessage.

2.1 Gestione dei Device interrupt tramite SSI

L'ssi si prende carico di generare l'interrupt a seguito di una richiesta DoIO, a questo punto il gestore degli interrupt invia direttamente al processo richiedente, bloccato sul device, il messaggio con lo status dell'operazione appena completata. Abbiamo evitato di usare l'ssi come tramite per la risposta del device per semplicità d'implementazione e di esecuzione.

3 Gestione delle liste per i device

Ogni device ha una lista dedicata. Queste liste sono riunite in un singolo vettore di 50 elementi: 48 dispositivi ordinati secondo le priorità delle interrupt lines, più una lista per l'Interval Timer e una general purpose per i processi che aspettano un messaggio oppure che sono appena stati sbloccati da un device.

3.1 Interval timer

Il tempo caricato sull'Interval Timer ad ogni ciclo viene calcolato in base alla seguente formula:

$$timeIT = PSECOND - (currTime \% PSECOND) \quad (1)$$

Dove PSECOND è la durata dell'interval timer e currTime è il valore del TOD. In questo modo, il timer viene impostato facendo riferimento al momento di inizio del TOD, permettendo di non accumulare eventuali ritardi causati dalla gestione dell'interrupt.

3.2 Device di I/O

Le liste dei processi bloccati sui device di I/O sono memorizzate in un vettore la cui mappatura tra indice e device segue la seguente formula:

$$i = \frac{devReg_{address} - devReg_{start_address}}{devReg_{size}} \quad (2)$$

Dove:

- $devReg_{address}$ è il parametro che indica l'indirizzo del device register associato al device per cui si vuole trovare il processo.
- $devReg_{start_address} = 0x10000054$ è l'indirizzo da cui iniziano i device register.
- $DevReg_{size} = 0x10$ è la dimensione dei device register (4 word).

Quindi al primo indice del vettore si trova la lista di processi associata al device di I/O con priorità maggiore (disco 1) e all'ultimo quella con priorità minore (trasmissione terminale 7).

4 Gestione del tempo di CPU

Il calcolo del tempo di CPU per ciascun processo utilizza, come da specifiche, il meccanismo di temporizzazione fornito dal TOD. La politica di accumulo prevede di contabilizzare, per ciascun processo, il tempo di esecuzione effettivo e il tempo impiegato per eseguire le system call (poiché sono direttamente invocate dal processo). Non viene invece considerato il tempo di gestione degli interrupt, durante il quale viene disabilitato il PLT, in quanto non sono causate in modo diretto dal processo.

5 Gestione dei pid

I pid vengono generati in modo incrementale a partire da 1. Il pid 1 è riservato all'ssi. Al momento non esistono controlli sull'incremento dei pid, quindi è teoricamente possibile arrivare a pid molto grandi ma per questa fase del progetto non abbiamo ritenuto necessario gestirli diversamente.