

I processi Unix

Ciascun processo Unix può evolvere in stato utente o in stato kernel. Il passaggio dall'uno all'altro stato avviene mediante SVC.

Poiché il processo può andare in attesa di un evento nell'uno o nell'altro stato, sono previste aree di salvataggio distinte nei due stati.

Il descrittore di un processo Unix è costituito da due strutture

- *entry della tabella dei processi*
- *struttura di utente*

La suddivisione è fatta per ottimizzare il rilascio di memoria in occasione di operazioni di swap

Entry della tabella dei processi

Contiene informazioni relative a:

Parametri di schedulazione: priorità, tempo di CPU recentemente speso, tempo di attesa recente.

Immagine di memoria: puntatori alle aree che contengono codice, dati e stack o tabella delle pagine. Se testo condiviso puntatore all'entry della tabella dei testi condivisi. Posizione su disco se il processo è stato swappato.

Segnali: maschera dei segnali. Un segnale può: essere ignorato, eliminare il processo che lo riceve, attivare una procedura.

Varie: stato del processo, evento che attende, tempo di allarme, pid, pid del padre, userid, groupid.

Struttura dell'utente

Contiene informazioni relative a:

Registri di macchina

Stato della chiamata di sistema: chiamata corrente compresi i parametri e risultato.

Tabella dei descrittori di file: tabella che permette di risalire all'i-node di un file aperto.

Addebito: tempo complessivo di cpu speso, tempo massimo di CPU spendibile, dimensione massima dello stack, numero massimo di pagine, etc.

Stack del kernel.

Scheduling

Lo scheduling si basa su una struttura di code a più livelli di priorità

La priorità di un processo in stato utente varia nel tempo mentre quella in stato kernel viene ridefinita in funzione dell'evento quando il processo va in attesa .

Il ricalcolo della priorità in stato utente tende a privilegiare i processi che in tempi recenti hanno effettuato CPU burst brevi.

Modalità di ricalcolo: Ogni secondo un daemon divide per due il contatore di utlizzi recente della CPU il cui valore definisce la nuova priorità e quindi la nuova posizione dei procesi nelle code. Il contatore di utilizzo viene incrementato di 1 ogni 20 msec quando il processo è running.

Gestione della memoria: lo swapping

Unix utilizza lo swapping per consentire che i processi possano non risiedere in memoria RAM in alcune fasi. Lo swapping è fondamentale nei sistemi Unix a partizioni.

Un processo viene swappato su disco se:

- chiede memoria per un processo figlio (fork)
- cerca di espandere il proprio segmento dati;
- il suo stack supera lo spazio riservato.

e la richiesta non può essere soddisfatta.

Lo swapping è realizzato mediante il daemon *swapper* che è presente indipendentemente se la memoria è gestita a partizioni o a memoria virtuale (pagine).

Gestione della memoria: lo swapping

Lo swapper interviene ogni pochi secondi per effettuare lo swap-in dei processi su disco pronti ad essere inseriti nelle code dei ready a partire da quello residente su disco da più tempo.

Il criterio utilizzato per liberare memoria è lo swap out di processi a partire da quelli meno prioritari e residenti da più tempo considerando prima i bloccati e poi i ready.

Un processo in memoria da meno di 2 sec. non viene swappato

Lo swapper si arresta quando non ci sono più processi pronti su disco o non è più possibile effettuare lo swap-out.

La memoria virtuale

La gestione della memoria virtuale si basa sulla paginazione.

La memoria è suddivisa in frame: lo stato di ogni frame è descritto da un entry della *core map* che è una tabella residente in memoria. La tabella è lunga 1 kbyte e una entry occupa 16 byte.

Una entry permette di sapere: se la pagina è libera, l'indirizzo del blocco di disco dove viene salvata/ripristinata nel paging, dove logicamente si colloca in un segmento, il tipo di segmento. Contiene inoltre informazioni sul suo stato.

La gestione della memoria virtuale è a pagine libere con algoritmo di sostituzione globale seconda chance.

La memoria virtuale

La disponibilità di frame liberi è controllata dal *pagedaemon* che viene attivato ogni 250 msec.

Il daemon controlla se la disponibilità di frame liberi è minore di un valore di riferimento *rif1*. Se vero, la riporta al valore *rif1* o al valore di riferimento *rif2* (dipende dalle versioni di Unix).

Per liberare le pagine applica l'algoritmo seconda chance percorrendo con due o un puntatore la coda circolare dei bit di riferimento associati ai frame.

Utilizza il primo puntatore per azzerare il bit se alto, il primo e il secondo puntatore, se esiste, per individuare il frame da liberare se basso (dipende dalle versioni di Unix).

La memoria virtuale

Se la frequenza di paginazione è troppo alta interviene lo swapper che libera tutti i frame occupati da processi non attivi da più di 20 sec. iniziando da quelli inattivi da maggior tempo.

Lo swapper poi periodicamente riassegna frame a processi swappati che sono pronti per essere schedulati. Vengono caricate solo la tabella utente e la tabella delle pagine.