

1. Cosa sono ed a cosa servono le chiamate di sistema (system call)? Si illustri brevemente il loro funzionamento.

Risposta: Le chiamate di sistema (system call) permettono ai programmi utente di richiamare i servizi del sistema operativo: sono solitamente disponibili come speciali istruzioni assembler o come delle funzioni nei linguaggi che supportano direttamente la programmazione di sistema (ad esempio, il C). Esistono vari tipi di chiamate di sistema relative al controllo di processi, alla gestione dei file e dei dispositivi, all'ottenimento di informazioni di sistema, alle comunicazioni. Come detto prima, l'invocazione di una chiamata di sistema serve ad ottenere un servizio dal sistema operativo; ciò viene fatto passando dal cosiddetto *user mode* al *kernel mode* per mezzo dell'istruzione speciale **TRAP**. Infatti, il codice relativo ai servizi del sistema operativo è eseguibile soltanto in kernel mode per ragioni di sicurezza. Una volta terminato il compito relativo alla particolare chiamata di sistema invocata, il controllo ritorna al processo chiamante passando dal kernel mode allo user mode.

2. In coda ready arrivano i processi P_1, P_2, P_3, P_4 , con CPU burst e istanti di arrivo specificati in tabella:

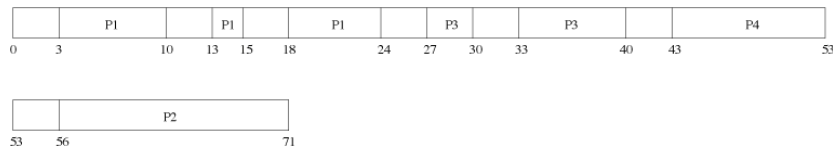
	arrivo	burst
P_1	0	15ms
P_2	10	15ms
P_3	15	10ms
P_4	30	10ms

Si consideri un tempo di latenza pari a 3ms e un algoritmo di scheduling SRTF. Si determini per i processi P_1, P_2, P_3, P_4 della tabella sopra:

- i) il diagramma di GANTT relativo all'esecuzione dei quattro processi;
- ii) il tempo di attesa medio;
- iii) il tempo di turnaround medio.

Risposta: Considerando un tempo di latenza pari a 3ms e un algoritmo di scheduling SRTF, abbiamo quanto segue:

- i) il diagramma di GANTT relativo all'esecuzione dei quattro processi è il seguente:



- ii) il tempo di attesa medio è $\frac{9+(56-10)+(27-15+3)+(43-30)}{4} = 20,75ms$,
- iii) il tempo di turnaround medio è: $\frac{24+(71-10)+(40-15)+(53-30)}{4} = 33,25ms$.

3. (a) Che cosa si intende con il termine *thrashing*? Che conseguenze può avere?

- (b) Descrivere il meccanismo del copy-on-write. Quando si applica e che vantaggio comporta?
- (c) Supponendo di avere un sistema con quattro frame e sette pagine, adottando una politica di rimpiazzamento LRU, quanti page fault si verificheranno con la reference string seguente?

3 4 0 1 0 2 4 5 1 2 6

(Si assuma che i quattro frame siano inizialmente vuoti.)

Risposta:

- (a) Quando la memoria fisica libera (e quindi il numero di frame liberi) è insufficiente a contenere il working set corrente di un processo, quest'ultimo comincerà presto a generare parecchi page fault, rallentando considerevolmente la propria velocità d'esecuzione. Quando parecchi processi cominciano ad andare in thrashing, ovvero a spendere più tempo per la paginazione che per l'esecuzione, il sistema operativo potrebbe erroneamente essere indotto a dedurre che sia necessario aumentare il grado di multiprogrammazione (dato che la CPU rimane per la maggior parte del tempo inattiva a causa dell'intensa attività di I/O). In questo modo vengono avviati nuovi processi che però, a causa della mancanza di frame liberi, cominceranno a loro volta ad andare in thrashing; in breve le prestazioni del sistema collassano fino ad indurre l'operatore a dover terminare forzatamente alcuni processi.
- (b) Il meccanismo del copy-on-write consente di condividere sia il codice che i dati a due processi. Ciò risulta utile in particolare in ambiente Unix in seguito ad una `fork` dato che sia il processo padre che il figlio inizialmente condividono sia il codice che i dati. Ciò avviene facendo in modo che le page table dei due processi puntino alle stesse pagine in memoria (in questo modo non è necessario duplicare le pagine dei dati al momento della `fork`). Tuttavia le pagine contenenti i dati vengono marcate come pagine di sola lettura. Quando uno dei due processi tenta di modificare una pagina dati, avviene una violazione della protezione di sola lettura e si solleva una trap al sistema operativo. La routine di gestione della trap a questo punto copia effettivamente la pagina ed aggiorna le page table in modo che ogni processo punti correttamente alla propria copia (viene anche tolto il flag di sola lettura ovviamente). In questo modo la dispendiosa attività di duplicare le pagine dei dati viene ritardata fino al primo tentativo di modifica da parte di uno dei processi coinvolti, facendo in modo che la `fork` venga eseguita molto velocemente.

(c) Simuliamo il funzionamento di LRU:

	3	4	0	1	0	2	4	5	1	2	6
		3	4	0	1	0	2	4	5	1	2
			3	4	4	1	0	2	4	5	1
				3	3	4	1	0	2	4	5
						3	3	1	0	0	4
								3	3	3	0
											3
	P	P	P	P		P		P	P		P

Si verificano quindi otto page fault.

4. Si spieghi brevemente cos'è il driver delle interruzioni e quali sono i passi che esegue durante il suo funzionamento.

Risposta: Il driver delle interruzioni è un componente fondamentale del software di un sistema operativo nei sistemi time-sharing e con I/O interrupt driven; infatti esso permette di gestire le interruzioni generate dai dispositivi di I/O, attivando le opportune routine di servizio. I passi eseguiti dal driver delle interruzioni sono i seguenti:

- salvare i registri della CPU,
 - impostare un contesto per la procedura di servizio (inizializzare TLB, MMU, stack ecc.),
 - inviare un segnale di *acknowledge* al controllore degli interrupt (per avere interrupt annidati),
 - copiare la copia dei registri nel PCB,
 - eseguire la procedura di servizio che accede al dispositivo,
 - eventualmente, cambiare lo stato a un processo in attesa (e chiamare lo scheduler di breve termine),
 - organizzare un contesto (TLB, MMU ecc.) per il processo successivo,
 - caricare i registri del nuovo processo dal suo PCB,
 - continuare l'esecuzione del processo selezionato.
5. (a) Si illustri la differenza fra architetture UMA e NUMA. I sistemi di calcolo appartenenti a queste due categorie sono strettamente accoppiati o debolmente accoppiati?
- (b) Cosa sono i cluster (o multicomputer)? Sono considerati dei sistemi strettamente accoppiati o debolmente accoppiati?

Risposta:

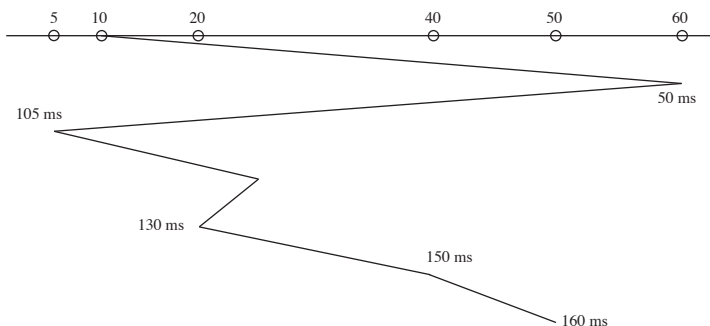
- (a) Un'architettura UMA (Uniform Memory Access) rappresenta un sistema che gode della proprietà che il tempo di accesso a qualunque locazione di memoria è sempre lo stesso (indipendentemente dal fatto che si faccia riferimento a memoria locale o di un altro nodo del sistema). Invece un sistema NUMA (Non Uniform Memory Access) è caratterizzato dal fatto che l'accesso alla memoria locale è più veloce (da 2 a 15 volte) dell'accesso alla memoria remota. Ciò consente

di ottenere sistemi scalabili con un gran numero di CPU. I sistemi di calcolo appartenenti a queste due categorie sono strettamente accoppiati.

- (b) Un cluster (o multicomputer) è un sistema di calcolatori strettamente accoppiati, ma senza memoria condivisa. La comunicazione avviene tramite il paradigma dello scambio di messaggi su linee dedicate ad alta velocità. I vari nodi sono costituiti da CPU, RAM, una scheda di rete ad alta velocità ed in alcuni casi un disco. Il resto delle periferiche sono condivise. Tutti i nodi solitamente sono situati all'interno di una stanza (spesso in armadi appositi) ed ognuno fa girare la propria copia dello stesso sistema operativo.
6. Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 10; lo spostamento ad una traccia adiacente richiede 1 ms. Al driver di tale disco arrivano richieste per i cilindri 60, 5, 40, 50, 20, rispettivamente agli istanti 0 ms, 30 ms, 100 ms, 110 ms, 125 ms. Si trascuri il tempo di latenza.
- (a) In quale ordine vengono servite le richieste?
- (b) Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le quattro richieste in oggetto?

Risposta:

- (a) Le richieste vengono soddisfatte nell'ordine: 60, 5, 20, 40, 50, come risulta dal seguente diagramma:



- (b) Il tempo di attesa medio per le cinque richieste in oggetto è
- $$\frac{(50-0)+(105-30)+(130-125)+(150-100)+(160-110)}{5} = \frac{50+75+5+50+50}{5} = \frac{230}{5} = 46 \text{ ms.}$$

Il punteggio attribuito ai quesiti è il seguente: 3, 5, 2, 2, 3, 3, 3, 3, 2, 2, 3, 2 (totale: 33).