

SISTEMI OPERATIVI

2 febbraio 2012

Cognome: _____ Nome: _____
Matricola: _____

1. Ricordate che non potete usare calcolatrici o materiale didattico.
2. Ricordate che potete consegnare al massimo tre prove scritte per anno accademico.

ESERCIZI RELATIVI ALLA PARTE DI TEORIA DEL CORSO

ESERCIZIO 1 (5 punti)

In un sistema operativo che adotta uno scheduling con diritto di prelazione, quattro processi arrivano al tempo indicato e consumano la quantità di CPU indicata nella tabella sottostante)

Processo	T. di arrivo	Burst
P1	0	12
P2	2	7
P3	5	3
P4	9	2

- a)
Qual è il waiting time medio migliore (ossia ottimale) che potrebbe essere ottenuto per lo scheduling dei quattro processi della tabella? RIPORTATE IL DIAGRAMMA DI GANTT USATO PER IL CALCOLO. (lasciate pure i risultati numerici sotto forma di frazione, e indicate quali assunzioni fate)

Diagramma di GANT, assumendo come algoritmo di scheduling SJF preemptive:

(0)....P1 ... (2) P2....(5)....P3....(8)....P2....(9)....P4...(11)....P2...(14)...P1....(24)

Waiting time medio:

$$P1 = (24 - 0) - 12 = 12;$$

$$P2 = (14 - 2) - 7 = 5;$$

$$P3 = (8 - 5) - 3 = 0;$$

$$P4 = (11 - 9) - 2 = 0;$$

$$\text{waiting time medio} = 17/4$$

- b)
Riportate lo pseudocodice che descrive la corretta implementazione dell'operazione di WAIT, e dite che funzione ha la system call usata nel codice.

Si vedano i lucidi della sezione 6.5.2.

- c)
All'interno di un sistema operativo, un certo processo P è correntemente in stato di "Ready to Run", e si sa che, una volta acquisita la CPU, non dovrà più rilasciarla volontariamente prima di aver terminato la propria esecuzione (in altre parole, non dovrà più eseguire operazioni di I/O, di sincronizzazione o di comunicazione con altri processi).

Quale/quali, tra gli algoritmi di scheduling **FCFS**, **SJF preemptive**, **SJF non-preemptive**, **round robin** garantisce/garantiscono che il processo P riuscirà a portare a termine la propria computazione? (motivate la vostra risposta, assumendo che SJF possa effettivamente essere implementato)

FCFS, e **round robin**. Infatti, nel caso di SJF (preemptive e non) potrebbe sempre arrivare in coda di ready un processo che deve usare la CPU per un tempo minore di quanto rimane da eseguire a P.

ESERCIZIO 2 (5 punti)

Si consideri un sistema in cui in una tabella delle pagine di un processo l'indice più grande usabile nella tabella delle pagine di quel processo può essere 3FFF. Un indirizzo fisico del sistema è scritto su 24 bit, e la RAM è suddivisa in 2000 (esadecimale) frame.

a)
Quanto è grande, in megabyte, lo spazio di indirizzamento logico del sistema (esplicitate i calcoli che fate)?

$2000(\text{esadecimale}) = 2^{13}$, per cui un numero di frame è scritto su 13 bit, e la dimensione di un frame, e quindi di una pagina, è di 2^{11} byte ($24 - 13 = 11$). Poiché il numero più grande di una pagina è 3FFF, ci possono essere al massimo 2^{14} pagine, e lo spazio di indirizzamento logico è di $2^{14} \times 2^{11}$ byte (pari a circa 32 megabyte).

b)
Quali informazioni conterrà ciascuna entry di una tabella delle pagine di questo sistema, se il sistema usa l'algoritmo di rimpiazzamento della seconda chance?

Il numero del frame che contiene la pagina corrispondente, il bit di validità della pagina, il reference bit.

c)
Nel caso non si verificano mai page fault, qual è, in nanosecondi, il tempo medio di accesso in RAM del sistema se viene usato un TLB con un tempo di accesso di 5 nanosecondi, un hit-ratio del 95% e un tempo di accesso in RAM di 0,08 microsecondi? (è sufficiente riportare l'espressione aritmetica che fornisce il risultato finale)

$$T_{\text{medio}} = 0,95 * (80+5) + 0,05 * (2*80 + 5) \text{ nanosecondi}$$

d)
Elencate tre vantaggi dell'uso delle librerie dinamiche

Possono essere condivise tra più processi, per cui occupano meno spazio in RAM.
Vengono caricate in RAM solo alla chiamata di una funzione, per cui i processi partono più velocemente.
Possono essere aggiornate senza dover ricompilare i programmi che le usano

e)
Commentate la seguente affermazione: in un qualsiasi sistema operativo che implementi la memoria virtuale, l'effettivo tempo di esecuzione di un programma può dipendere pesantemente dal modo con cui il programma accede ai propri dati.

L'affermazione è vera. Ad esempio, l'accesso per colonne ad array memorizzati per riga può produrre un elevatissimo numero di page fault, e quindi un forte rallentamento nell'esecuzione del programma.

ESERCIZIO 3 (4 punti)

a) Descrivete (se preferite usando un disegno) il metodo di allocazione dello spazio su disco adottato dal sistema operativo Unix.

Si vedano i lucidi della sezione 11.4.3.

b) Scegliete una dimensione per i blocchi dell'hard disk e il numero di bit usati per scrivere il numero di un blocco, e calcolate di conseguenza la dimensione massima che può avere un file di quel sistema (è sufficiente riportare l'espressione che permette di calcolare la dimensione del file).

Blocco: 1024 byte, puntatore a blocco: 4 byte.

Dimesione massima: $10 \times 1k + 256 \times 1k + 256^2 \times 1k + 256^3 \times 1k$

c) come è fatta la struttura interna di una qualsiasi cartella unix, e quali entry sono sempre presenti in qualsiasi cartella unix? (se preferite usate un disegno)

È un array in cui ciascuna entry è formata dal nome di un file e dal numero di un index-node che contiene tutte le informazioni del file stesso. Qualsiasi cartella unix contiene sempre almeno le due entry "." E ".."

d) Cosa sono gli hard link unix, e perché non sono ammessi hard link tra directory?

Si vedano i lucidi della sezione sui link unix.