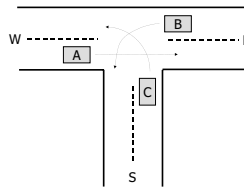


Università degli studi di Verona
Corso di Laurea in Informatica/Tecnologie dell'Informazione
Sistemi Operativi
16 Luglio 2003

1. Si consideri la situazione che si verifica ad un incrocio tra tre strade come indicato in figura.



Tre auto A , B , e C devono procedere nelle direzioni indicate, rispettando le regole della precedenza a destra. E , W , S rappresentano le tre direzioni (East, West, South).

Considerando A , B , e C come tre processi, scrivere uno pseudocodice di una soluzione basata su semafori che coordini la situazione indicata. Si indichino con $\text{GoEast}()$ (per A), $\text{GoSouth}()$ (per B), e $\text{GoWest}()$ (per C) le operazioni corrispondenti alle tre direzioni delle auto.

SUGGERIMENTI:

- Si utilizzi un semaforo binario per modellare il fatto che un'automobile prende possesso dell'incrocio (quindi tre semafori in totale). A seconda della direzione, un'auto dovrà prendere possesso di uno o più semafori.
- Le tre auto non arrivano necessariamente all'incrocio contemporaneamente. La soluzione deve essere generale in modo da prevedere il caso in cui una macchina arrivi prima delle altre. [8 punti]

-
2. Si consideri il seguente insieme di processi:

Processo	Burst	Tempo di Arrivo
1	3	0
2	4	1
3	8	1
4	1	5

Si mostri l'esecuzione dei processi usando uno schedatore a tre livelli (tre code Q_1, Q_2, Q_3) con feedback. Q_1 è la coda a più alta priorità, Q_3 quella a priorità più bassa. Q_1 e Q_2 utilizzano un algoritmo round-robin, con quanto $q_1 = 1$ e $q_2 = 2$, rispettivamente. Q_3 usa un algoritmo FIFO. Più precisamente, i processi entrano nella coda Q_1 , e se non terminano entro q_1 unità di tempo passano a Q_2 ; qui, se non terminano entro q_2 unità di tempo passano a Q_3 . Il dispatcher preleva sempre dalle code in ordine decrescente di priorità. Disegnare il diagramma temporale per le tre code, e calcolare il tempo di **attesa** per ogni processo). [5 punti]

-
3. Si descriva l'algoritmo di scheduling utilizzato da UNIX. [6 punti]

-
4. Un programma durante la sua esecuzione scandisce ripetutamente un vettore che occupa 1000 pagine. Ogni scansione parte dalla pagina 0 e continua fino alla 999 (e così ciclicamente). Al processo vengono assegnati 100 frame.

In questa configurazione, nè l'algoritmo FIFO nè l'algoritmo LRU offre prestazioni accettabili. Dopo aver valutato il numero (o il tasso) di page fault risultanti dall'utilizzo dell'algoritmo LRU, si discuta un algoritmo che sulla stringa di riferimento di questo esempio ottenga i risultati ottenibili con l'algoritmo ottimo (ma che sia ovviamente implementabile). Si valutino le prestazioni di questo algoritmo.

[7 punti]

-
5. Si consideri un sistema che utilizza paginazione su domanda, per il quale sono state misurate le seguenti utilizzazioni temporali (cioè la % di utilizzazione nel tempo):

Utilizzazione della CPU	20%
Disco (per paginazione)	97%
Altri dispositivi di I/O	5%

Quali delle seguente variazioni nel sistema possono eventualmente migliorare l'**utilizzazione della CPU**? Si risponda chiaramente (SI/NO) e si commenti successivamente in dettaglio ogni risposta:

- (a) Installare una CPU più veloce;
- (b) Installare un disco più grande;
- (c) Aumentare il grado di multiprogrammazione;
- (d) Diminuire il grado di multiprogrammazione;
- (e) Installare dispositivi di I/O più veloci;

[5 punti]