

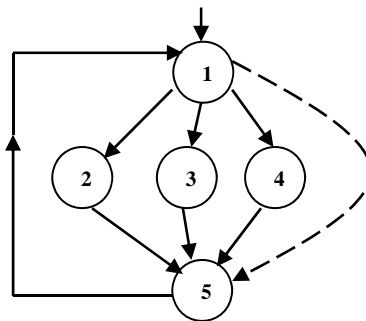
Sistemi Operativi

18.02.2011

1. Che cosa produce su video questo programma, il cui eseguibile è e1, quando viene dato il comando `for((i=0;i<=3;i++)) do e1 $i; done ? Perché?`

```
#include <stdio.h>
#include <unistd.h>
int main(int argc, char **argv){
char *v[6] = {"/bin/ls", "/bin/echo", "/bin/ls", "e1", "3", NULL};
printf("pronto ad eseguire \n");
execv (v[atoi(argv[1])], &v[atoi(argv[1])]);
printf("eseguito \n");
}
```

2. Realizzare il grafo di precedenza indicato in figura utilizzando il numero minimo di semafori e tenendo conto che i processi sono ciclici, inoltre ad ogni ciclo P4 può essere eseguito o meno come suggerito dall'arco tratteggiato1-5.



3. Si supponga di avere a disposizione, oltre alle primitive **WAIT** e **SIGNAL**, anche una primitiva semaforica **TRY_WAIT** che permette di testare il suo contatore e di bloccare il processo chiamante solo se il contatore decrementato diventa negativo. Utilizzando queste primitive realizzare il codice di due produttori e di un consumatore. I produttori inseriscono i loro messaggi dopo un'attesa casuale ciascuno nel proprio buffer (**Buf[0]** o **Buf[1]**). Il consumatore deve prelevare prima da **Buf[0]**, se contiene almeno un messaggio, oppure da **Buf[1]** se contiene almeno un messaggio. Ovviamente si vuole evitare che il consumatore attenda su **Buf[0]** se c'è già un messaggio disponibile su **Buf[1]**.

4. Realizzare uno script **AWK** che, dato un file di input nel linguaggio X, stampi a video un output secondo le seguenti regole:

- ogni riga che comincia con ``/*`` e che termina con ``*/`` non deve essere stampata;
- ogni riga del tipo ``#define stringa1 stringa2`` deve essere sostituita da ``stringa1 = stringa2``;
- ogni riga del tipo ``#define stringa`` deve invece rimanere tale e quale;
- ogni altra riga che non corrisponde a una delle regole suddette deve essere ristampata inalterata.

Inoltre, al termine, lo script deve stampare un messaggio in cui si riporti il numero di funzioni, identificate dalle righe che cominciano con uno dei seguenti identificatori (**int**, **float**, **char**, **void**) seguito da una stringa composta solo da lettere minuscole e/o maiuscole seguita da ``{`` (esempio, ``int CalcolaRisultato{``).

5. Realizzare uno script **bash** che modifichi la variabile di ambiente **LD_LIBRARY_PATH** rimuovendo tutte le directory che contengono la stringa ``.lib64``. Si può usare **awk SOLO** per dividere la stringa contenuta nella variabile **LD_LIBRARY_PATH**, di cui segue un esempio:

```
/usr/lib:/usr/lib64:/usr/local/lib:/usr/local/lib64
```

6. Descrivere i passi con cui la shell **bash** espande i seguenti comandi (riportare esplicitamente tutti i passi per ogni comando):

```
variabile="{1,2,3}"
find . -name "*.txt" -exec a.sh `echo $variabile == {1,2,3}` \{\} \;
```