

Ex. 1	
Ex. 2	
Ex. 3	
Ex. 4	
Ex. 5	
Ex. 6	
Tot.	

Sistemi Operativi

Compito d'esame

28 Gennaio 2016

Matricola _____ Cognome _____ Nome _____

Docente: Quer Sterpone

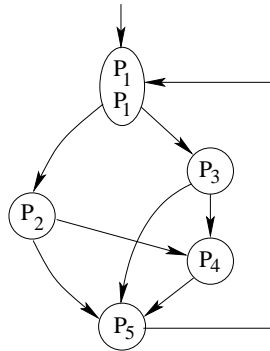
L'unico materiale consultabile durante la prova scritta consiste nei tre formulari predisposti dal docente. Riportare i passaggi principali. L'ordine sarà oggetto di valutazione.

Durata della prova: 100 minuti.

1. Si riporti il control flow graph e l'albero di generazione dei processi ottenuto dall'esecuzione del seguente tratto di codice C. Si indichi inoltre che cosa esso produce su video e per quale motivo.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
int main () {
    int i, j=-1;
    for (i=0; i<5; i++) {
        printf ("i=%d\n", i);
        if (!fork()) {
            for (j=2; j>0; j--) {
                fork();
                execlp ("echo", "i", "j", (char *) 0);
            }
        } else {
            printf ("j=%d \n", j);
        }
    }
    return (1);
}
```

2. Dato il seguente grafo di precedenza, realizzarlo utilizzando il **minimo** numero possibile di semafori. I processi rappresentati devono essere processi ciclici (con corpo del tipo `while(1)`). Utilizzare le primitive `init`, `signal`, `wait` e `destroy`. Indicare gli eventuali archi superflui e riportare il corpo dei processi (P_1, \dots, P_5) e l'inizializzazione dei semafori. Si noti che P_1 deve effettuare due iterazioni in sequenza del proprio ciclo `while` per ogni iterazione globale sull'intero grafo di precedenza.



3. Con riferimento alle Sezioni Critiche, si indichino i requisiti che ogni soluzione deve soddisfare e quali classi di soluzioni sono proponibili (con le rispettive caratteristiche e diversità).

Si faccia quindi riferimento alla sincronizzazione di processi mediante la procedura `swap`. Se ne illustrino le principali caratteristiche, riportandone il codice e il relativo protocollo di utilizzo.

4. Scrivere uno script BASH (senza l'utilizzo di AWK) che riceve tre parametri:

```
n cmd dir
```

in cui `n` indica un valore numerico intero, `cmd` è una stringa che rappresenta un comando (singolo) di shell e `dir` è il nome di un direttorio.

Lo script deve eseguire il comando `cmd` sui primi `n` file presenti nel direttorio `dir`. Si supponga che `n` sia sempre inferiore del numero di file presenti nel direttorio e che i file vadano selezionati in ordine di dimensione decrescente.

Suggerimento: si ricorda che nell'output del comando `ls -l`

```
-rw-rw-r-- 1 quer quer 42729 Jan 25 11:33 file1.txt
```

```
-rw-rw-r-- 1 quer quer 226662 Jan 25 11:33 exam.ps
```

```
...
```

i campi sono separati da una tabulazione e la dimensione del file è indicata nel quinto campo.

5. La base dati relativa a un esame è costituita da due file.

Il primo file riporta le generalità degli studenti, con formato:

```
cognome nome numeroMatricola
```

Il secondo file memorizza il risultato ottenuto da tutti gli studenti in tutti gli esami superati:

```
numeroMatricola nomeEsame voto
```

Scrivere un script AWK in grado di ricevere sulla riga di comando 5 parametri:

```
nomeFile1 nomeFile2 numeroMatricola x sogliaVoto
```

dove nomeFile1 e nomeFile2 sono file con il formato precedentemente descritto e x è un operatore relazionale, ovvero una tra le stringhe (<, ==, >). Lo script deve visualizzare (a video) l'elenco di tutti gli esami sostenuti dallo studente di matricola indicata che soddisfano la condizione relazionale "voto x sogliaVoto". Il formato di uscita deve essere del tipo:

```
cognome nome nomeEsame voto
```

6. Si consideri il seguente insieme di processi:

Processo	Tempo arrivo	Burst Time	Priorità
P ₁	0	2	5
P ₂	1	8	1
P ₃	2	1	4
P ₄	3	6	2
P ₅	4	4	3

Rappresentare mediante diagramma di Gantt l'esecuzione di tali processi utilizzando gli algoritmi di scheduling PS (Priority Scheduling), RR (Round Robin) e SRTF (Shortest Remaining Time First).

La priorità maggiore è associata al valore di priorità inferiore. Per la strategia Round Robin si utilizzi un quantum temporale di 2 unità di tempo.