Caselle riservate

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

Sistemi Operativi

Compito d'esame

28 Gennaio 2013

Versione A

Matricola _____ Cognome _____ Nome _____

Docente: ◯ Laface ◯ Quer

Non si possono consultare testi, appunti o calcolatrici. Riportare i passaggi principali. L'ordine sarà oggetto di valutazione.
Durata della prova: 60 **minuti.**

1. Si riporti l'albero di generazione dei processi e si indichi che cosa produce su video il seguente programma e per quale motivo.

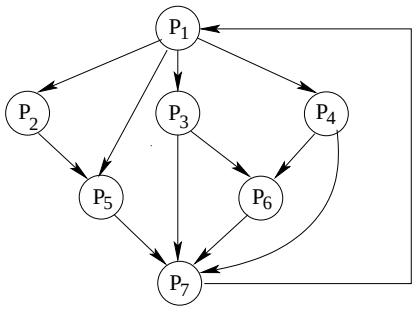
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(){
  pid_t pid;
  int i;
  for (i=1; i<=3; i++){
    switch (i) {
      case 1: fork(); break;
      case 2: pid=fork(); if (pid!=0) system ("echo case 2"); break;
      case 3: execlp ("echo", "myPgrm", "case 3", NULL); break;
    }
  }
  return (0);
}
```

2. Si illustri il problema dei *Readers e Writers* riportandone la soluzione per il caso di precedenza ai Readers mediante semafori. Si indichi la funzione dei vari semafori motivandone l'utilizzo.

3. Si illustri l'*algoritmo del banchiere*. Analizzando l'esempio successivo (con processi $(P_0, \ldots, P_4)$ e risorsa R) si indichi se lo stato è sicuro o non sicuro e si riporti la sequenza sicura o non sicura.

Processo	Fine	Assegnate R	Massimo R	Necessità R	Disponibilità R
P_0	No	2	7		2
P_1	No	2	3		
P_2	No	2	8		
P_3	No	0	3		
P_4	No	1	5		

4. 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)`). Si utilizzino le primitive `init`, `signal` e `wait`. Riportare il corpo dei processi $(P_1, \ldots, P_7)$ e l'inizializzazione dei semafori.



5. Realizzare uno script bash che riceva come unico argomento un file di testo. Lo script deve:
   - effettuare una copia del file in un file con lo stesso nome ma con estensione `xyx`
   - modificare il file originario come segue:
     - aggiungere all'inizio di ogni riga il numero di parole della riga e il numero di righe totali del file
     - ordinare le righe in ordine crescente in base al numero di parole.

   Non si ricorra all'utilizzo di AWK.

6. Un file contiene un testo di lunghezza indefinita ma senza caratteri di interpunzione. Scrivere uno script AWK che, ricevuto il nome di tale file sulla riga di comando, visualizzi su standard output l'istogramma a barre del numero di occorrenze di tutte le stringhe presenti nel file di lunghezza esattamente uguale a 5 caratteri e contenenti almeno due vocali qualsiasi tra 'a', 'e', 'i', 'o', 'u'.

Esempio

File di ingresso	Output prodotto
testo di esempio che contiene molte parole	`testo` ###
con 5 caratteri e almeno 2 vocali testo	`molte` ####
barre molte molte molte barre di testo	`barre` ##

Caselle riservate

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

Sistemi Operativi

Compito d'esame

28 Gennaio 2013

Versione B

Matricola _____ Cognome _____ Nome _____

Docente: ◯ Laface ◯ Quer

Non si possono consultare testi, appunti o calcolatrici. Riportare i passaggi principali. L'ordine sarà oggetto di valutazione.
Durata della prova: 60 **minuti.**

1. Si riporti l'albero di generazione dei processi e si indichi che cosa produce su video il seguente programma e per quale motivo.

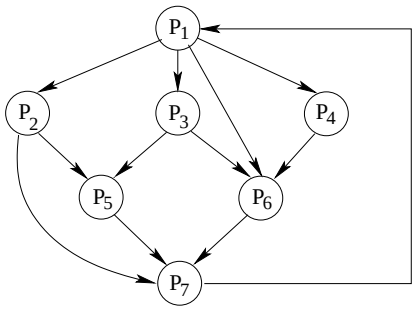
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(){
  pid_t pid;
  int i;
  for (i=3; i>=1; i--){
    switch (i) {
      case 1: execlp ("echo", "myPgrm", "case 1", NULL); break;
      case 2: fork(); break;
      case 3: pid=fork(); if (pid!=0) system ("echo case 3"); break;
    }
  }
  return (0);
}
```

2. Si illustri il problema dei *Produttore e Consumatore* riportandone la soluzione per il caso di un solo produttore e di tre consumatori. Si indichi la funzione dei vari semafori motivandone l'utilizzo.

3. Si illustri l'*algoritmo del banchiere*. Analizzando l'esempio successivo (con processi $(P_0, \ldots, P_4)$ e risorsa R) si indichi se lo stato è sicuro o non sicuro e si riporti la sequenza sicura o non sicura.

Processo	Fine	Assegnate R	Massimo R	Necessità R	Disponibilità R
P_0	No	2	11		2
P_1	No	2	3		
P_2	No	3	8		
P_3	No	0	3		
P_4	No	1	5		

4. 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)`). Si utilizzino le primitive `init`, `signal` e `wait`. Riportare il corpo dei processi $(P_1, \ldots, P_7)$ e l'inizializzazione dei semafori.



5. Uno script bash riceve sulla riga di comando il nome di tre direttori. Lo script deve visualizzare (a video) l'elenco dei nomi dei file contenuti nel primo direttorio che contengono la stringa `main` e l'elenco dei file che non la contengono. Inoltre deve copiare il primo insieme di file nel secondo direttorio e il secondo insieme di file nel terzo direttorio.

   Se il secondo e il terzo direttorio non esistono, lo script deve crearli; in caso contrario deve cancellare tutti i file in essi contenuti prima dell'esecuzione dello script. Lo script controlli inoltre il corretto passaggio dei parametri.

   Non si ricorra all'utilizzo di AWK.

6. Due file di testo `a.txt` e `b.txt` dovrebbero contenere le stesse parole anche se non nello stesso ordine. Implementare uno script AWK che verifichi se tutte le parole presenti nel primo file sono presenti anche nel secondo file con lo stesso numero di occorrenze. Visualizzare le parole che non rispettano questa condizione.