

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

# Sistemi Operativi

## Compito d'esame

06 Febbraio 2017

Matricola \_\_\_\_\_ Cognome \_\_\_\_\_ Nome \_\_\_\_\_

Docente:  Quer  Sterpone

**Non si possono consultare testi, appunti o calcolatrici a parte i formulari distribuiti dal docente. Riportare i passaggi principali. L'ordine sarà oggetto di valutazione.**

**Durata della prova: 100 minuti.**

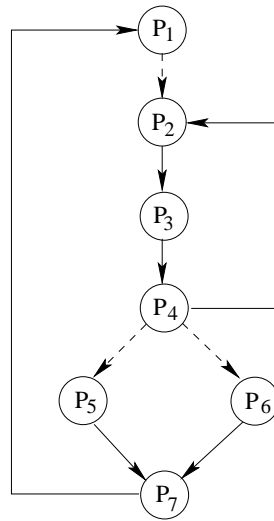
1. Si riporti il grafo di controllo del flusso e l'albero di generazione dei processi a seguito dell'esecuzione del seguente tratto di codice C. Si supponga che il programma venga eseguito con un unico parametro, il valore intero 3, sulla riga di comando. Si indichi inoltre che cosa esso produce su video e per quale motivo.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/wait.h>
5
6 int main (int argc, char *argv[]) {
7     char str[20];
8     int n = atoi (argv[1]);
9
10    if (n>=0 && fork()) {
11        if (fork(>0) {
12            sprintf (str, "%d", n);
13            execlp ("echo", "echo", "n=", str, (char *) 0);
14            if (fork(>0) {
15                execlp (argv[0], argv[0], n-1, (char *) 0);
16            }
17        } else {
18            sprintf (str, "%s %d", argv[0], n-1);
19            printf ("%s\n", str);
20            if (fork(>0) {
21                system (str);
22            }
23        }
24    }
25
26    return (1);
27 }

```

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)`). Inoltre il ciclo interno (processi 2, 3 e 4) deve essere eseguito 3 volte per ogni iterazione del ciclo esterno: la prima esecuzione incomincia una volta terminato  $P_1$ , mentre le successive due incominciano una volta terminato  $P_4$ ; inoltre, i processi  $P_5$  e  $P_6$  devono essere eseguiti solo una volta terminata la terza iterazione. Utilizzare le primitive `init`, `signal`, `wait` e `destroy`. Indicare gli eventuali archi superflui, riportare il corpo dei processi ( $P_1, \dots, P_7$ ) e l'inizializzazione dei semafori.



3. Con riferimento alle soluzioni software e hardware al problema della sincronizzazione si descrivano la strategia di Peterson e quella basata sulla procedura di `swap`. Per entrambe le strategie, si riporti il codice e se ne illustrino le funzionalità, il protocollo di utilizzo, i pregi e i difetti.

4. Due file contengono esclusivamente numeri interi, separati da un o più spazi e disposti con formato libero, ovvero in numero indefinito su ciascuna riga del file. Scrivere uno script BASH in grado di ricevere il nome di tali file sulla riga di comando, di verificare il loro corretto passaggio e l'esistenza dei file, e di visualizzare solo i numeri che **non** compaiono in entrambi i file.

5. Scrivere uno script AWK in grado di manipolare l'output di un riconoscitore vocale secondo le seguenti specifiche.

Il riconoscitore vocale genera un file con le seguenti caratteristiche:

```
pronounced word : one           utterance n. : 1
recognized word  : one           prob. : -0.15530174E+02
pronounced word : one           utterance n. : 2
recognized word  : one           prob. : -0.92723360E+01
pronounced word : one           utterance n. : 3
recognized word  : one           prob. : -0.95529728E+01
pronounced word : one           utterance n. : 4
recognized word  : eight        prob. : -0.11542629E+02
pronounced word : one           utterance n. : 5
recognized word  : one           prob. : -0.10308277E+02
pronounced word : two           utterance n. : 6
recognized word  : two           prob. : -0.15609604E+02
pronounced word : two           utterance n. : 7
recognized word  : two           prob. : -0.81284819E+01
pronounced word : two           utterance n. : 8
recognized word  : two           prob. : -0.92986431E+01
pronounced word : two           utterance n. : 9
recognized word  : zero         prob. : -0.12929784E+02
```

in cui diverse parole pronunciate possono essere riconosciute correttamente o meno con una determinata probabilità. Il nome di tale file viene ricevuto dallo script sulla riga di comando.

Lo script deve quindi parsificare tale file e produrre in uscita la cosiddetta “matrice di confusione” con il seguente formato:

	one	zero	two	eight	tot	percent_correct
one	4	0	0	1	5	80.00%
zero	0	0	0	0	0	
two	0	1	3	0	4	75.00%
eight	0	0	0	0	0	

In essa le parole pronunciate e/o riconosciute sono riportate lungo le righe e le colonne e, in base al numero di riconoscimenti indicati negli elementi stessi della matrice, vengono calcolate le percentuali di correttezza (e.g.,  $4/5 = 0.8$  ovvero 80%,  $3/4 = 0.75$  ovvero 75%, etc.).

6. Si supponga che un disco rigido sia costituito da 25 blocchi, che i blocchi liberi siano indicati con 0 e quelli occupati con 1, e che la situazione iniziale del disco sia la seguente:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	0	1	1	0	0	0	0	0	1	0	0	1	1	1	1	0	0	0	0	1	0	0	1

Indicare le principali caratteristiche delle metodologie di allocazione di file contigua, concatenata, FAT e indicizzata. Utilizzando tali tipi di allocazione e il disco precedentemente descritto, indicare come possono essere allocati in sequenza i seguenti file: `File1`, `File2` e `File3` ciascuno di dimensione uguale a 4 blocchi.