

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

Sistemi Operativi

Compito d'esame

06 Febbraio 2015

Matricola _____ Cognome _____ Nome _____

Non si possono consultare testi, appunti o calcolatrici. Riportare i passaggi principali. L'ordine sarà oggetto di valutazione.

Durata della prova: 100 minuti.

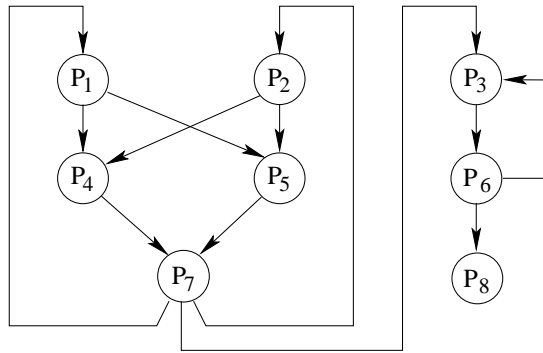
1. Si riporti l'albero di generazione dei processi a seguito dell'esecuzione del seguente tratto di codice C. Si supponga che il programma venga eseguito con un unico parametro, il valore intero 3, sulla riga di comando. Si indichi inoltre che cosa esso produce su video e per quale motivo.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>
int main (int argc, char *argv[]) {
    int i;
    char str[50];
    i = atoi (argv[1]);
    printf ("0 - run with i=%d\n", i); fflush (stdout);
    if (i<=0) exit (0);
    if (fork () > 0) {
        sprintf (str, "echo 1 - run with i=%d", i);
        system (str);
        sprintf (str, "%d", i-1);
        execlp (argv[0], argv[0], str, NULL);
    } else {
        sprintf (str, "echo 2 - run with i=%d", i);
        system (str);
        sprintf (str, "%d", i-2);
        execlp (argv[0], argv[0], str, NULL);
    }
    exit (0);
}
```

2. Si indichi sinteticamente il significato dei seguenti termini e/o concetti.

Funzioni e system call	
Process Control Block	
Processo init	
Processo zombie	
Funzione rientrante e race condition	
Half-duplex pipe	
Context switching	
Thread kernel e thread utente	

3. 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_8) e l'inizializzazione dei semafori.



4. Un file, creato dal comando “`ls -laR`”, è stato successivamente semplificato in modo da ottenere il seguente file:

```
total 24
drwxrwxr-x 6 quer quer 4096 Jan 31 16:38 .
drwxr-xr-x 4 quer quer 4096 Feb 10 13:30 ..
drwxrwxr-x 3 quer quer 4096 Feb 10 17:08 current
-rwxrwxr-x 1 quer quer 9037 Jan 30 23:24 pgrm
-rw-rw-rw- 1 quer quer 8881 Jan 30 23:49 pgrm.c
...
total 1116
drwxrwxr-x 3 quer quer 4096 Feb 11 17:08 .
...
```

In cui la parola chiave `total` serve esclusivamente a iniziare l’elenco di un nuovo sotto-direttorio, i campi di ciascun elenco sono separati da uno spazio singolo e il quinto campo indica lo spazio occupato dalla directory entry.

Scrivere uno script BASH in grado di ricevere il nome di un file con tale formato sulla riga di comando e di visualizzare una qualche informazione (il nome, la dimensione, tutta la riga, etc., a scelta) sui file che occupano più spazio di tutti gli altri file inclusi nel direttorio a cui essi stessi appartengono. Ad esempio, nel primo direttorio tale file è `pgrm`.

5. Scrivere un script AWK in grado di generare l'elenco delle citazioni bibliografiche riportate in un testo secondo le seguenti specifiche.

Un testo contiene un numero illimitato di citazioni bibliografiche con il seguente formato:

```
(id) citazione (ID)
```

in cui una stringa qualsiasi (ma con tutti caratteri alfabetici minuscoli e racchiusa tra parentesi tonde) rappresenta l'inizio della citazione, e la stessa stringa (ma con tutti caratteri alfabetici maiuscoli) rappresenta la sua terminazione. Ciascuna citazione è in generale costituita da più parole, è inclusa interamente in un'unica riga, e può comparire più volte all'interno del testo (tutte le citazioni racchiuse dallo stesso identificatore sono identiche). Le parentesi tonde non sono utilizzare per altri scopi all'interno del file.

Il file di ingresso così formattato va ricopiato in un file di uscita, in cui ogni citazione viene sostituita all'interno del testo con un numero intero crescente (a partire da 1) e va quindi inserita in fondo al testo stesso con formato:

```
(numero crescente) citazione
```

A numero uguale corrisponde ovviamente citazione uguale.

Il nome del file di ingresso e del file di uscita vanno ricevuti dallo script sulla riga di comando o in alternativa assegnati quali variabili sulla riga di comando stessa.

Il seguente esempio riporta a sinistra un possibile file di ingresso e a destra il corrispondente file di uscita.

```
...
... (soa) A. Silbershatz, ... (SOA) ...
...
(sob) W. R. Stevens, ... (SOB)
... (soa) ... (SOA) ...
...
... (sob) ... (SOB) ...

...
... (1) ....
...
(2)
... (1) ...
...
... (2) ...
BIBLIOGRAFIA
(1) A. Silbershatz, ...
(2) W. R. Stevens, ...
```

6. Per i candidati iscritti al corso nell'anno accademico 2014–2015.

Si consideri il seguente insieme di processi:

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

Rappresentare mediante diagramma di Gantt l'esecuzione di tali processi utilizzando gli algoritmi di scheduling "First Come First Served" senza prelazione e "Shortest Job First" senza e con prelazione.

Calcolare il tempo di completamento per ciascun processo, nonché il tempo di attesa medio. Confrontare i tre algoritmi in base a quest'ultimo criterio, motivando il risultato.

Per i candidati iscritti al corso negli anni accademici 2012–2013 o 2013–2014.

Chiarire il significato dei seguenti termini: *directory entry*, *hard-link*, *soft-link*. Riportare i comandi per creare hard- e soft-link. Riportare un esempio di gestione e di conteggio degli hard-link nel caso della creazione di un direttorio quale sotto-direttorio di un direttorio dato. Rappresentare la filosofia generale e la struttura del file-system mediante ausili grafici opportuni.