

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

# Sistemi Operativi

## Compito d'esame

12 Settembre 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     setbuf (stdout, 0);
11     if (n<=0)
12         return (1);
13     printf ("run with n=%d\n", n);
14     if (fork()>0) {
15         sprintf (str, "%d", n-1);
16         printf ("exec\n");
17         execlp (argv[0], argv[0], str, (char *) 0);
18     } else {
19         sprintf (str, "%s %d", argv[0], n-2);
20         printf ("system\n");
21         system (str);
22     }
23     wait ((int *) 0);
24
25     return (1);
26 }
```

2. Si descriva l'utilizzo dei segnali nel sistema operativo UNIX/Linux indicandone i principali vantaggi e svantaggi.

Si implementi inoltre il seguente programma. Un processo genera due figli e quindi attende il loro completamento. Il primo figlio rimane in attesa di segnali provenienti dal secondo figlio e visualizza (su standard output) un messaggio diverso a seconda del segnale ricevuto dal fratello. Il secondo figlio:

- (a) Genera un numero casuale intero  $r$  incluso nell'intervallo  $[1, 3]$ .
- (b) Se  $r = 1$ , attende 1 secondo, uccide il fratello inviandogli il segnale `SIGKILL` e termina a sua volta.
- (c) Se  $r = 2$  attende 2 secondi e invia al fratello il segnale `SIGUSR1` e ritorna la punto (a).
- (d) Se  $r = 3$  attende 3 secondi e invia al fratello il segnale `SIGUSR2` e ritorna la punto (a).

Si utilizzi l'espressione `((int)(rand()%3)+1)` per generare un numero casuale nell'intervallo  $[1, 3]$ .

3.

Si indichino le differenze tra una funzione e una system call.

Si indichi che cosa è il processo “init”.

Si indichi che cosa è un processo “zombie” e si riporti il codice C necessario per generarne uno.

Si indichi che cosa significa avere permessi `-rw-rw-r--` e come tali permessi si possono modificare.

Si specifichi che cosa significa eseguire il programma `p` con il comando: `p >& f &`

4. Il comando di shell `echo $PATH` visualizza un elenco del seguente tipo:

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Scrivere uno script BASH che, una volta ricevuta una stringa sulla riga di comando, sia in grado di eliminare dalla stringa generata dal comando precedente tutti i riferimenti a sotto-alberi della stringa specificata e di visualizzare la stringa risultante.

Per esempio, nel caso del path precedente, se il programma ricevesse la stringa `local` dovrebbe visualizzare la seguente stringa:

```
/usr/sbin:/usr/bin:/sbin:/bin
```

5. Il comando di shell `cal july 2017` (oppure il comando analogo `cal 7 2017`) fornisce la seguente stampa:

```
      July 2017
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

Scrivere un script AWK in grado di ricevere sulla riga di comando una stringa (nell'insieme "Su", "Mo", "Tu", "We", "Th", "Fr" e "Sa") e il numero di un anno (e.g., 2017) e di visualizzare il nome di tutti i mesi di quell'anno il cui primo giorno è quello indicato.

Per esempio nel 2017 l'unico mese che inizia per "Mo" è "May".

6. Il comando `ls -la` sul direttorio di lavoro produce il seguente risultato:

```
total 20
drwxrwxr-x  4 quer quer 4096 Jan 25 13:43 .
drwx----- 10 quer quer 4096 Jan 25 13:39 ..
drwxrwxr-x  4 quer quer 4096 Jan 25 13:42 dir1
drwxrwxr-x  2 quer quer 4096 Jan 25 13:39 dir2
-rw-rw-r--  1 quer quer  44 Jan 25 13:43 file
```

Si illustri che cosa indica il secondo valore riportato su ciascuna riga, come esso viene generato e come viene modificato a seguito di ciascuno dei seguenti comandi:

```
mkdir dir1/a
mkdir dir1/b
mkdir dir1/a/c
ln file l1
ln file l2
```

In particolare, si illustrino i concetti di hard-link, soft-link e inode nel sistema operativo UNIX/Linux.