

Nome Cognome, Numero Matricola:

## Grader per Esame Sistemi Operativi - Modulo 2 - Canale 2 - 06.07.2012

Enrico Tronci

Dipartimento di Informatica, Università di Roma "La Sapienza", Via Salaria 113, Roma, Italy

tronci@di.uniroma1.it,

<http://mclab.di.uniroma1.it>

July 6, 2012

**Durata Esame: 20 min**

Rispondere al maggior numero di domande. Le risposte errate valgono punti -1. Non risposta vale punti 0. Le risposte esatte valgono punti 2.

Rispondere sul foglio risposte mettendo una croce **a penna** (non a matita) sulla risposta scelta. Se si decide di non rispondere mettere una croce sullo 0. Per ogni domanda deve esserci esattamente una croce sul foglio risposte. Per minimizzare errori di copiatura, scrivere direttamente sul foglio risposte a penna. Se si decide di cambiare risposta è sufficiente scrivere una N sulla risposta da cancellare e segnare una croce sulla risposta desiderata.

**Q 1** Nel sistema operativo UNIX, l'istruzione `kill:`

- (a) **OK** Può mandare un segnale ad un processo.
- (b) Può solo terminare l'esecuzione di un processo.
- (c) Può solo terminare la sessione corrente.

**Q 2** Nel sistema operativo UNIX, l'istruzione `diff pippo pluto` ha la seguente semantica:

- (a) Stampa su stdout le righe del file `texttppippo` che non sono nel file `pluto`.
- (b) **OK** Stampa su stdout le differenze tra i files `pippo` e `pluto`.
- (c) Stampa su stdout le righe del file `textttpluto` che non sono nel file `pippo`.

**Q 3** Nel sistema operativo UNIX, l'istruzione `rm -fr pippo` ha la seguente semantica:

- (a) Rimuove ricorsivamente il contenuto della directory `pippo` (che alla fine è vuota) senza rimuovere la directory `pippo`.
- (b) **OK** Rimuove ricorsivamente il contenuto della directory `pippo` e la directory `pippo` stessa.
- (c) Rimuove solo i files contenuti nella directory `pippo`.

**Q 4** Si considerino i seguenti file attributes nel tipico formato UNIX:

```
-rw-r--r-- 1 enrico tronci 278 Jul 5 13:12  
pippo.tex
```

Lo user del file è:

(a) **tronci**

(b) Non definito dalla lista di attributi di cui sopra.

(c) **OK** **enrico**

**Q 5** Nel sistema operativo UNIX, l'istruzione per spostarsi dalla directory corrente alla directory `TARGET` è:

(a) `mv TARGET`

(b) **OK** `cd TARGET`

(c) `cp TARGET`

**Q 6** L'effetto dell'istruzione `cat pippo` è:

(a) Stampare la stringa `pippo` su stdout.

(b) **OK** Stampare il contenuto del file `pippo` su stdout.

(c) Stampare la lista dei files il cui nome inizia con `pippo`.

**Q 7** Nel sistema operativo UNIX, l'istruzione `sort pippo` ha la seguente semantica:

(a) **OK** Ordina le righe del file `pippo` in ordine lessicografico crescente e stampa su stdout il risultato.

(b) Ordina le parole del file `pippo` in ordine lessicografico crescente e stampa su stdout il risultato.

(c) Stampa su stdout in ordine lessicografico crescente il contenuto della directory `pippo`.

**Q 8** L'effetto dell'istruzione  
`pwd`  
è:

- (a) **OK** Stampare la directory corrente.
- (b) Stampare la lista dei processi in esecuzione.
- (c) Stampare la lista delle directories contenute nella directory corrente.

**Q 9** Nel sistema operativo UNIX, l'istruzione  
`find . -name "*.tex"`  
ha la seguente semantica:

- (a) **OK** Stampa su stdout, ricorsivamente a partire dalla directory corrente, tutti i files il cui nome termina con la stringa `.tex`.
- (b) Stampa su stdout tutti i files della directory corrente il cui nome termina con la stringa `.tex`.
- (c) Stampa su stdout tutti i files della home (dello user) il cui nome termina con la stringa `.tex`.

**Q 10** Un processo P diventa zombie quando:

- (a) Il parent process di P termina senza mai eseguire una `wait` e poi termina P.
- (b) Prima termina P e poi termina il parent process di P senza mai eseguire una `wait`.
- (c) **OK** P termina, il parent process di P non termina e non esegue mai una `wait`.

**Q 11** Nel sistema operativo UNIX, l'istruzione  
`chmod +x pippo`  
ha la seguente semantica:

- (a) Sposta il file `pippo` nella directory `x`.
- (b) Rinomina il file `pippo` aggiungendo una `x` alla fine. Il nuovo nome è: `pippox`.
- (c) **OK** Rende il file `pippo` eseguibile.

**Q 12** La semantica dell'istruzione  
`ls -l *.qry | wc | gawk '{print $1}'`  
è la seguente:

- (a) Per ogni file, nella directory corrente, il cui nome termina con `.qry`, stampa su stdout il nome del file e la sua size in bytes.
- (b) Per ogni file, nella directory corrente, il cui nome termina con `.qry`, stampa su stdout il nome del file ed il suo numero di righe.
- (c) **OK** Stampa su stdout il numero di files della directory corrente il cui nome termina con `.qry`.

**Q 13** L'effetto dell'istruzione  
`exec echo 3`  
nella shell di login è:

- (a) Stampare 3 su stdout e restituire il prompt.

- (b) **OK** Stampare 3 su stdout e terminare la sessione.
- (c) Stampare `echo 3` su stdout e restituire il prompt.

**Q 14** Nel sistema operativo UNIX, l'istruzione  
`echo prova > pippo`  
ha la seguente semantica:

- (a) Copia nel file `pippo` il contenuto del file `prova`. Il precedente contenuto di `pippo` è sovrascritto.
- (b) **OK** Scrive nel file `pippo` la stringa `prova`. Il precedente contenuto di `pippo` è sovrascritto.
- (c) Appende al file `pippo` la stringa `prova`. Il precedente contenuto di `pippo` è conservato.

**Q 15** Si consideri il seguente script

```
verb=sing
echo I like ${verb}
```

Una possibile esecuzione dello script di cui sopra è:

- (a) I like `${verb}`
- (b) **OK**  
I like sing
- (c) I like verb

**Q 16** Una pipe è uno pseudo file usato per:

- (a) Comunicazione tra processi su computers distinti.
- (b) Comunicazione tra un computer e la stampante su rete.
- (c) **OK** Comunicazione tra processi sullo stesso computer.

**Q 17** Nel sistema operativo UNIX, l'istruzione  
`cp -r pippo pluto`  
ha la seguente semantica:

- (a) Copia il file `pippo` nel file `pluto`.
- (b) Copia nella directory `pluto` i files nella directory `pippo`.
- (c) **OK** Copia ricorsivamente nella directory `pluto` il contenuto della directory `pippo`.

**Q 18** Si considerino i seguenti file attributes nel tipico formato UNIX:

```
-rw-r--r-x 1 enrico tronci 278 Jul 5 13:12
pippo.tex
```

Lo user del file ha i seguenti permessi:

- (a) Lettura, ma non scrittura od esecuzione.
- (b) **OK** Lettura e scrittura, ma non esecuzione.
- (c) Lettura ed esecuzione, ma non scrittura.

**Q 19** Nel sistema operativo UNIX, l'istruzione `gawk '($1 == "Stampa") {print $0}' pippo` ha la seguente semantica:

- (a) Stampa su stdout tutte le occorrenze della parola `Stampa` nel file `pippo`.
- (b) Stampa su stdout tutte le righe del file `pippo` che contengono la parola `Stampa`.
- (c) **OK** Stampa su stdout tutte le righe del file `pippo` che contengono la parola `Stampa` come prima parola.

**Q 20** Una socket di tipo `AF_INET` può essere usata per la comunicazione tra processi:

- (a) Su macchine ovunque su Internet, ma non sulla stessa macchina.
- (b) Solo sulla stessa macchina.
- (c) **OK** Su macchine ovunque su Internet, anche sulla stessa macchina.

**Q 21** L'istruzione:  
`signal(SIGINT, myhandler);`  
ha il seguente effetto:

- (a) Manda al processo `myhandler` il segnale `SIGINT`.
- (b) **OK** Installa `myhandler` come handler per il segnale `SIGINT`.
- (c) Manda al processo che la esegue il segnale `SIGINT` e poi esegue `myhandler`.

**Q 22** L'effetto dell'istruzione  
`man cmd`  
è:

- (a) Eseguire `cmd`.
- (b) **OK** Stampare su stdout una descrizione del comando `cmd`.
- (c) Stampare su stdout `cmd`.

**Q 23** L'effetto dell'istruzione  
`eval 'echo pwd'`  
nella shell di login è:

- (a) Stampare `pwd` su stdout.
- (b) Stampare `echo pwd` su stdout.
- (c) **OK** Stampare il path della directory di lavoro su stdout.

**Q 24** Nel sistema operativo UNIX, l'istruzione  
`egrep sck pippo`  
ha la seguente semantica:

- (a) **OK** Mostra su stdout tutte le righe del file `pippo` che contengono il pattern `sck`.
- (b) Mostra su stdout la prima riga del file `pippo` che contiene il pattern `sck`.

- (c) Cerca il file `sck` nella directory `pippo`.

**Q 25** Nel sistema operativo UNIX, l'istruzione  
`echo prova >> pippo`  
ha la seguente semantica:

- (a) Appende al file `pippo` il contenuto del file `prova`. Il precedente contenuto di `pippo` è conservato.
- (b) Cancella il precedente contenuto del file `pippo` e scrive in `pippo` la stringa `prova`.
- (c) **OK** Appende al file `pippo` la stringa `prova`. Il precedente contenuto di `pippo` è conservato.

**Q 26** Nel sistema operativo UNIX, l'istruzione per spostare un file `pippo` dalla directory corrente alla directory `TARGET` è:

- (a) **OK** `mv pippo TARGET`
- (b) `cp pippo TARGET`
- (c) `xpath pippo TARGET`

**Q 27** Nel sistema operativo UNIX, l'istruzione per stampare su stdout i files di una directory è:

- (a) `dir`
- (b) `pwd`
- (c) **OK** `ls`

**Q 28** Il file *file descriptor* ritornato dall' `system call` `open()` è di tipo

- (a) `FILE *`
- (b) **OK**  
`int`
- (c) `char *`

**Q 29** Nel sistema operativo Unix il "path name" di un file viene considerato relativo alla directory di lavoro quando:

- (a) Il path name inizia con `..`.
- (b) Il path name inizia con un carattere alfanumerico.
- (c) **OK** Il path name NON inizia con `/`.

**Q 30** Nel sistema operativo UNIX, il comando `crontab` serve per:

- (a) Schedulare l'esecuzione di comandi che vengono eseguiti una sola volta.
- (b) **OK** Schedulare l'esecuzione di comandi che vengono eseguiti periodicamente.
- (c) Stampare su stdout in ordine cronologico i comandi eseguiti dall'utente.

**Q 31** Nel sistema operativo UNIX, l'istruzione `rmdir pippo` ha la seguente semantica:

- (a) **OK** Rimuove la directory `pippo` se e' vuota, altrimenti ritorna errore.
- (b) Rimuove la directory `pippo` in ogni caso.
- (c) Cancella il contenuto della directory `pippo` (che alla fine é vuota) senza rimuovere la directory `pippo`

**Q 32** Nel sistema operativo UNIX, l'istruzione `uniq pippo` ha la seguente semantica:

- (a) Stampa su stdout il file `pippo` collassando tutte le parole adiacenti ripetute in un unica parola.
- (b) Stampa su stdout il file `pippo` collassando tutte i caratteri adiacenti ripetuti in un unico carattere.
- (c) **OK** Stampa su stdout il file `pippo` collassando tutte le righe adiacenti ripetute in un unica riga.