

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

Sistemi Operativi

Compito d'esame

29 Giugno 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 descrivano le caratteristiche principali delle *pipe* per la comunicazione e la sincronizzazione tra processi. Se ne illustri l'utilizzo realizzando il seguente programma in linguaggio C. Un processo P genera un figlio F_1 che a sua volta genera un figlio F_2 . F_2 legge da tastiera un intero, la trasmette a F_1 su una prima pipe e termina. F_1 trasforma il numero nel suo opposto e lo trasmette a P su una seconda pipe e termina. P visualizza il numero ricevuto su standard output e termina.

2. Con riferimento allo scheletro di una shell UNIX/Linux si scriva un programma, in linguaggio C, in grado di:
- leggere da tastiera dei comandi Linux con eventuali parametri sulla riga di comando (ad esempio “cp file1 file2”, “ls -laR”, etc.)
 - eseguire tali comandi in background, procedendo (in foreground) alla lettura del comando successivo.
 - terminare il procedimento, ovvero la lettura di nuovi comandi Linux una volta ricevuto da tastiera il comando “end”.

Si osservi che, a scelta del candidato, la stringa “end” può essere utilizzata anche per delimitare la fine di ciascun comando letto da tastiera.

3. Si riportino quattro possibili soluzioni software al problema delle sezioni critiche con due processi denominati P_i e P_j . Si motivi l'erroneità oppure la correttezza di ciascuna soluzione.

4. Si riportino i comandi UNIX per effettuare le operazioni indicate, utilizzando eventuali ridirezioni e pipe:
- Nel direttorio `/home/fooo` cercare i file con il nome che inizia con il carattere `L` e estensione `.txt`. In questi file ricercare al presenza della stringa `laib`. Visualizzare il nome del file e l'intera riga in cui tale stringa viene rintracciata.
 - Trovare tutti i file di estensione `.txt` nel direttorio `/home` memorizzati tra il livello di profondità 3 (incluso) e il livello di profondità 5 (incluso) dell'albero dei direttori e che siano leggibili. Di questi modificare il proprietario in `ugo`.
 - Per ogni file di estensione `.txt` presente nel direttorio corrente ricavare il nome e il numero di caratteri presenti nel file. L'elenco venga ordinato in base al numero di caratteri in ordine numerico inverso e memorizzato nel file `stat.txt`.
 - Fare un esempio di utilizzo di ciascuno dei tre filtri `cut`, `tr`, `uniq`, spiegando l'effetto di ciascun comando.
 - Una applicazione C è formata dai file `main.c`, `f1.c`, `f2.c` e `main.h`. Scrivere un Makefile con due target. Il primo sia in grado di compilare l'applicazione denominando l'eseguibile `myapp`. Il secondo rimuova eventuali file temporanei e sposti l'eseguibile nel direttorio `/user/bin`.

5. Scrivere un script AWK in grado di effettuare sostituzioni multiple in un testo soddisfacendo alle specifiche successive.

Un primo file include su righe successive coppie di stringhe, nel formato:

```
stringa1 stringa2
```

Lo script deve sostituire ogni comparsa della `stringa1` con la `stringa2` in un file di ingresso, generando il corrispondente file di uscita. Si osservi che le stringhe possono degenerare in caratteri singoli e devono essere rintracciate e sostituite nel testo in ingresso anche come sotto-stringhe di stringhe di lunghezza maggiore. Si supponga inoltre che tutte le sostituzioni siano indipendenti e possano essere applicate in qualsiasi ordine. Lo script riceve tre stringhe sulla riga di comando. Tali stringhe individuano il file di conversione, quello di ingresso e quello di uscita da generare.

Esempio

Si supponga lo script venga eseguito come segue:

```
myScript file.txt inFile.txt outFile.txt
```

e che i file di nome `file.txt` e `inFile.txt` abbiano il contenuto di seguito indicato. Lo script deve generare il file `outFile.txt` riportato.

<code>file.txt</code>	<code>inFile.txt</code>	<code>outFile.txt</code>
awk Awk unix UNIX o O	ScriptawkInIngresso Contenente unix e UNIX e testo vario	ScriptAwkInIngressO COntenente UNIX e UNIX e testO variO

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

Indicare le principali caratteristiche delle tipologie di allocazione di file (contigua, concatenata, indicizzata). Che cosa si intende per FAT? Che cosa si intende per collegamento o link?

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

Si illustri la struttura di un file system UNIX con particolare riferimento alla memorizzazione di file e direttori. Che cosa si intende per collegamento o link?