

Sistemi Operativi T

Prova di laboratorio

15 luglio 2024

Si realizzi un file comandi unix con la seguente interfaccia:

```
esame dir M fout
```

dove:

- `dir` è il nome assoluto di una directory esistente nel file system
- `M` è un intero positivo
- `fout` è il nome assoluto di un file

Dopo aver effettuato gli opportuni controlli dei parametri in ingresso, il file comandi dovrà scandire ricorsivamente il sottoalbero individuato da `dir` e considerare solo il regular file di proprietà dell'utente che ha invocato lo script. Per tali file, dovrà contare il numero X di parole contenute nelle prime 3 righe (si vedano in proposito i comandi `head` e `wc`)

Per ogni sottodirectory trovata (inclusa la directory `dir` stessa) dovrà determinare la somma SUM dei valori di X per i file in essa contenuti e, qualora tale somma risulti maggiore di M passato come parametro (quindi se $SUM > M$), dovrà aggiungere una riga nel file `fout` passato come parametro.

Se `fout` dovesse già esistere al momento dell'invocazione dello script, si abbia cura di cancellarne il contenuto prima di iniziare a popolarlo.

Ogni riga del file `fout` deve avere il seguente formato:

`<absDirName> <SUM>`

Dove:

- `<absDirName>` è il nome assoluto della directory ispezionata e
- `<SUM>` è il valore della somma sopra descritta per la cartella `absDirName`

Sistemi Operativi T
Prova di laboratorio
14 giugno 2023
Tema A

Esercizio di Programmazione Shell (punti 10)

Si realizzi uno script bash che implementi la seguente interfaccia:

\$./esame IN OUT

dove:

- **IN** e **OUT** sono nomi assoluti di directory esistenti nel file system.

Si supponga che il sottoalbero individuato da **IN** contenga, disordinatamente, una serie di file, ciascuno relativo ad un prodotto *<prodX>* commercializzato da un'azienda. Ogni file porta il nome *<prodX>* del prodotto cui si riferisce e contiene come seconda parola della prima riga il nome *<fornY>* del fornitore di tale prodotto.

Lo script deve esplorare RICORSIVAMENTE la directory **IN** per ordinarne il contenuto. In particolare, si vuole che la cartella di destinazione **OUT** contenga i file raggruppati in sottocartelle per fornitore.

Per ogni file ordinario *<prodX>* trovato nella directory **IN**, deve:

1. Usare i comandi `head` e `cut` (oppure `head` e `awk`) per filtrare la seconda parola della prima riga (la quale riporta il nome del fornitore *<fornY>*).
2. Qualora non esista già, creare una cartella *<fornY>* in **OUT**
3. Copiare il file *<prodX>* dentro alla cartella *<fornY>*

Sistemi Operativi T
Prova di laboratorio
14 giugno 2023
Tema B

Esercizio di Programmazione Shell (punti 10)

Si realizzi uno script bash che implementi la seguente interfaccia:

\$./esame IN OUT

dove:

- **IN** e **OUT** sono nomi assoluti di directory esistenti nel file system.

Si supponga che il sottoalbero individuato da **IN** contenga, disordinatamente, una serie di file, ciascuno relativo ad una prenotazione *<prenotX>* in un hotel. Ogni file porta il nome *<prenotX>* della prenotazione (un codice alfanumerico univoco) e contiene come seconda parola della prima riga il nome *<clienteY>* del cliente che ha effettuato la prenotazione.

Lo script deve esplorare RICORSIVAMENTE la directory **IN** per ordinarne il contenuto. In particolare, si vuole che la cartella di destinazione **OUT** contenga i file raggruppati in sottocartelle per cliente.

Per ogni file ordinario *<prenotX>* trovato nella directory **IN**, deve:

4. Usare i comandi `head` e `cut` (oppure `head` e `awk`) per filtrare la seconda parola della prima riga (la quale riporta il nome del cliente *<clienteY>*).
5. Qualora non esista già, creare una cartella *<clienteY>* in **OUT**
6. Copiare il file *<prenotX>* dentro alla cartella *<clienteY>*

Sistemi Operativi T
Prova di laboratorio
14 giugno 2023
Tema C

Esercizio di Programmazione Shell (punti 10)

Si realizzi uno script bash che implementi la seguente interfaccia:

\$./esame INprod OUTprod

dove:

- **INprod** e **OUTprod** sono nomi assoluti di directory esistenti nel file system.

Si supponga che il sottoalbero individuato da **INprod** contenga, disordinatamente, una serie di file, ciascuno relativo ad un prodotto *<prodX>* commercializzato da un'azienda. Ogni file porta il nome *<prodX>* del prodotto cui si riferisce. Inoltre, l'utente proprietario del file *<prodX>* fornisce indicazione di quale sia l'addetto *<addZ>* di magazzino responsabile di gestire quel prodotto.

Lo script deve esplorare RICORSIVAMENTE la directory **INprod** per ordinarne il contenuto. In particolare, si vuole che la cartella di destinazione **OUTprod** contenga i file raggruppati in sottocartelle per username *<addZ>* dell'addetto responsabile.

Per ogni file ordinario *<prodX>* trovato nella directory **INprod**, deve:

1. Usare il comando `stat` (ed in particolare l'opzione `--format=%U`) per ricavare lo username dell'utente proprietario del file (indicante l'addetto *<addZ>* di magazzino responsabile di gestire quel prodotto).
2. Qualora non esista già, creare una cartella *<addZ>* in **OUTprod**
3. Copiare il file *<prodX>* dentro alla cartella *<addZ>*

Sistemi Operativi T
Prova di laboratorio
14 giugno 2023
Tema D

Esercizio di Programmazione Shell (punti 10)

Si realizzi uno script bash che implementi la seguente interfaccia:

\$./esame INpre OUTpre

dove:

- **INpre** e **OUTpre** sono nomi assoluti di directory esistenti nel file system.

Si supponga che il sottoalbero individuato da **INpre** contenga, disordinatamente, una serie di file, ciascuno relativo ad una prenotazione *<prenotX>* in un hotel. Ogni file porta il nome *<prenotX>* della prenotazione (un codice alfanumerico univoco). Inoltre, l'utente proprietario del file *<prenotX>* fornisce indicazione di quale sia il receptionist *<recZ>* responsabile di gestire quella prenotazione.

Lo script deve esplorare RICORSIVAMENTE la directory **INpre** per ordinarne il contenuto. In particolare, si vuole che la cartella di destinazione **OUTpre** contenga i file raggruppati in sottocartelle per username *<recZ>* del receptionist responsabile.

Per ogni file ordinario *<prenotX>* trovato nella directory **INpre**, deve:

4. Usare il comando `stat` (ed in particolare l'opzione `--format=%U`) per ricavare lo username dell'utente proprietario del file (indicante il receptionist *<recZ>* responsabile di gestire quella prenotazione).
5. Qualora non esista già, creare una cartella *<recZ>* in **OUTprod**
6. Copiare il file *<prenotX>* dentro alla cartella *<recZ>*

Sistemi Operativi T

Prova di laboratorio

11 luglio 2022

Si realizzi un file comandi unix con la seguente interfaccia:

esame stringa outDir dir1...dirN

dove:

- **stringa** è una stringa
- **dirOut** e **dir1...dirN** sono path assoluti di directory esistenti nel file system (N non è noto a priori).

Dopo aver effettuato gli opportuni controlli dei parametri in ingresso, il file comandi dovrà scandire ricorsivamente tutti i sottoalberi individuati da **dir1...dirN**.

Per ogni cartella (e sottocartella) trovata dovrà controllare se più della metà dei regular file in essa contenuti, presenta al suo interno almeno un'occorrenza di **stringa**.

Pertanto, lo script dovrà contare per ogni cartella:

- il numero X di file contenuti
- il numero Y di file che contengono almeno un'occorrenza di **stringa**

Se $Y > X/2$, lo script dovrà aggiungere una riga riepilogativa in un file di report situato in **outDir**.

La riga avrà il formato seguente:

<pathAbs> <Y>

Dove **<pathAbs>** è il nome assoluto del direttorio ispezionato e **<Y>** è il risultato del conteggio sopraindicato.

Il nome del file riepilogativo in **outDir** dovrà rispettare il formato seguente:

report.<PID>.out, dove **<PID>** è il pid del processo che ha eseguito lo script.

Sistemi Operativi T

Prova di laboratorio

29 giugno 2022

Esercizio di Programmazione bash Unix [punti 10]

Si realizzi un file comandi unix con la seguente interfaccia:

esame xUser yUser est dir

dove:

- **xUser**, **yUser** sono stringhe
- **est** è una stringa che deve iniziare per “.” e che rappresenta un’estensione di file
- **dir** è una directory esistente nel file system

Dopo aver effettuato gli opportuni controlli dei parametri in ingresso, il file comandi dovrà scandire ricorsivamente il sottoalbero **dir** e individuare in quali direttori esistono più file con estensione **est** di proprietà di **xUser** che file con estensione **est** di proprietà di **yUser**.

Pertanto, per ogni sottocartella trovata (inclusa la cartella **dir** stessa) lo script deve considerare solo i regular file con estensione **est** (i.e., il cui nome termina con **est**) e contare:

- il numero nX di file di proprietà dell’utente **xUser**
- il numero nY di file di proprietà dell’utente **yUser**

Per filtrare il nome dell’utente proprietario di un file si consiglia di utilizzare il comando **awk** visto a lezione.

Inoltre, lo script dovrà scrivere un resoconto dentro al file **report** situato nella HOME directory dell’utente che ha eseguito lo script. Qualora il file **report** non esistesse al momento del lancio, dovrà essere creato. Se esistente, il suo contenuto dovrà essere troncato.

Per ogni directory per cui $nX > nY$, lo script dovrà scrivere in **report** una riga del tipo: **<nomeAssDirectory> <Diff>**

Dove:

- **<nomeAssDirectory>** è il nome assoluto della directory il cui contenuto rispetta la specifica precedente ($nX > nY$)
- **<Diff>** è la differenza tra nX e nY

Sistemi Operativi T

Prova di laboratorio

13 Luglio 2021

1. Esercizio di programmazione shell con bash Linux [punti 10]

Si realizzi un file comandi unix con la seguente interfaccia:

esame ORIGINE LOG ERR LISTA1 LISTA2 ... LISTAN

dove:

- **ORIGINE** è il path assoluto di un file esistente che termina con estensione “.txt” su cui si hanno permessi di lettura.
- **LOG** è il path relativo o assoluto di un file esistente su cui si hanno permessi di scrittura.
- **ERR** è il path relativo o assoluto di un file che può o meno esistere già, su cui si hanno permessi di scrittura.
- **LISTA1...N** sono path assoluti di file su cui si hanno permessi di lettura che contengono al loro interno una serie di nomi di cartelle, uno per riga.

Il programma deve come prima cosa controllare la correttezza dei parametri.

Successivamente, per ogni file **LISTAi** ($i=1,..N$), deve leggerne il contenuto. Per ogni cartella del file, se la cartella esiste nella working directory e se si hanno i diritti adeguati, il programma deve copiare al suo interno il file **ORIGINE**. Poi deve scrivere in append al file **LOG** il contenuto della cartella.

Nel caso una delle cartelle non esista o non si abbiano i diritti necessari, il programma deve stampare gli errori sul file **ERR**, eventualmente sovrascrivendone una versione precedentemente esistente. La stampa dell'errore consiste nello scrivere il nome del file **LISTA** relativo, il nome della cartella e quale dei due casi di errore si è verificato (la cartella non esiste o non si hanno i diritti).

IMPORTANTE: il file di errore **ERR** non deve essere sovrascritto ogni volta che c'è un errore, va sovrascritta solo un'eventuale versione precedente all'avvio del programma!

Sistemi Operativi T

Prova di laboratorio

25 Giugno 2021

2. Esercizio di programmazione shell con bash Linux [punti 10]

Si realizzi un file comandi unix con la seguente interfaccia:

esame DIRIN STRING EXT FOUT

Dove:

- **DIRIN** è il nome di una cartella esistente nella working directory.
- **STRING** è una stringa
- **EXT** è una stringa di 4 caratteri, nella quale il primo elemento **deve essere** un punto (ovvero: il carattere ' . ')
- **FOUT** è il path assoluto di un file di testo esistente.

Dopo aver effettuato gli opportuni controlli sui parametri di ingresso, il processo deve analizzare ricorsivamente i sottoalberi del direttorio DIRIN. Per ogni sottoalbero, deve individuare tutti i file ordinari contenuti al suo interno, il cui nome termini con la stringa EXT.

Per ogni file di estensione EXT deve verificare che contenga al suo interno la stringa STRING. Nel caso la stringa sia presente, deve scrivere (in append) nel file FOUT:

- il nome assoluto del file
- la sua dimensione
- il nome utente del suo proprietario
- il numero di occorrenze di STRING trovate all'interno del file

in questo preciso ordine e separati dal carattere ":" (due punti)

Esempio di riga del file FOUT (se EXT=".txt")

/user/anna/doc/nomefile1.txt:56:utente:3

Sistemi Operativi T
Prova di laboratorio
30 Giugno 2020

2. Programmazione shell [punti 10]

Si realizzi un file comandi bash con la seguente interfaccia:

`esame fileToSearch S M dir1 ... dirN`

dove:

- `fileToSearch` è una stringa
- `S` e `M` sono interi positivi
- `dir1...dirN` sono nomi assoluti di directory esistenti nel file system (`N` non noto a priori).

Dopo aver effettuato gli opportuni controlli sui parametri di ingresso, lo script deve analizzare **RICORSIVAMENTE** per `M` volte a intervalli regolari di `S` secondi tutti i sottoalberi individuati da `dir1...dirN`, verificando la presenza di uno o più file comuni denominati `fileToSearch`.

Per ogni file trovato deve aggiungere una riga ad un file di log situato nella HOME directory dell'utente che ha invocato lo script.

Il nome del file di log deve avere il seguente formato:

`<PID><fileToSearch>.log`

Dove `<PID>` è il process idenfier del processo di monitoraggio in esecuzione (quello che sta eseguendo lo script `esame`), e `<fileToSearch>` è il nome del file passato come parametro.

Se un file con tale nome dovesse già essere presente nella home dell'utente, si abbia cura di cancellarne il contenuto prima di iniziare il monitoraggio.

Ogni riga del file di log deve invece avere il seguente formato:

`<date> :: We found <Nome assoluto di fileToSearch>`

Dove `<date>` è la data e ora corrente, così come restituita dall'apposito comando di shell.