

PROGETTO SISTEMI OPERATIVI 2017/1028

Elaborato 1: Shell

Consegna entro: 08 aprile 2018; (23:59)

Si vuole creare un programma per monitorare l'utilizzo di un server basato su bash script. Il programma di monitoraggio deve realizzare tre funzionalità:

1. Salvare la lista dei processi in esecuzione sul server.
2. Riportare informazioni di utilizzo del server.
3. Fornire un'interfaccia, basata su menu testuale, che permetta ad un amministratore di sistema l'utilizzo delle funzionalità 1 e 2.

Di seguito sono riportate le specifiche di ogni funzionalità. I requisiti funzionali non esplicitamente definiti, possono essere liberamente implementati purché siano comandi bash e non si richieda l'installazione di third-party software sui PC dei laboratorio.

Specifiche funzionalità 1:

La funzionalità che salva la lista dei processi in esecuzione sul server deve essere implementata con un script di nome *recorder.bash*. Lo script *recorder.bash* periodicamente (comando *sleep*) salva in un file csv *report_<date>_<time>.csv* la lista dei processi in esecuzione sul server. <date> e <time> sono rispettivamente la data (yyyymmdd) e l'ora (hhss) attuale di sistema al momento della lettura dei processi in esecuzione. Ogni riga del file csv riporta un processo in esecuzione. Per ogni processo, devono essere riportate le seguenti informazioni su file: {user, PID, command, start, time(durata)}. Esempio file csv generato:

```
user1;19;ps aux;12:23;0:00
user2;53;firefox;13:00;0:03
user3;99;vim main.c;13:22;1:00
user1;130;firefox;12:00;3:00
...
```

report_20170318_1230.csv

Specifiche funzionalità 2:

La funzionalità che fornisce informazioni di utilizzo del server deve essere implementata con un script di nome *query.bash*. Lo script *query.bash* stampa a video le seguenti informazioni:

1. Per ogni utente X, il numero dei processi in esecuzione appartenenti a X riportati nel csv file generato più recentemente.

Esempio:

```
Last report: report_20170318_1230.csv
user1: 2
user2: 1
user3: 1
....
```

2. Per un utente X fornito da riga di comando, il numero di processi in esecuzione appartenenti a X riportati in ogni csv file generato.

Esempio:

```
Inserire utente: user1
report: report_20170318_1230.csv processi: 2
report: report_20170318_1215.csv processi: 10
report: report_20170318_1200.csv processi: 15
report: report_20170318_1145.csv processi: 0
....
```

Specifiche funzionalità 3:

La funzionalità che fornisce l'interfaccia del programma di monitoraggio deve essere implementata con un script di nome *menu.bash*.

menu.bash visualizza sul terminale il seguente menu testuale:

```
Monitoraggio server. Comandi disponibili:
1)  Inizia monitoraggio
2)  Ferma monitoraggio
3)  Stampa info utilizzo
4)  Chiudi
Inserisci numero comando [1-4]:
```

Per ogni comando disponibile, le seguenti operazioni devono essere eseguite:

1. *Inizia monitoraggio*: lo script `recorder.bash` viene lanciato in modalità background. Se `recorder.bash` è già in esecuzione, allora nessuna operazione deve essere eseguita.
2. *Ferma monitoraggio*: lo script `recorder.bash` viene killato. Se `recorder.bash` non è in esecuzione, allora nessuna operazione deve essere eseguita.
3. *Stampa info utilizzo*: lo script `query.bash` viene eseguito.
4. *Chiudi*: lo script `menu.bash` termina.

Se il numero del comando inserito non è riconosciuto, il menu testuale deve essere visualizzato nuovamente sul terminale.

BONUS

Realizzare lo script `checkSpace.bash` per controllare lo spazio utilizzato da tutti i file csv generati dalla funzionalità 1 sul filesystem. Se lo spazio utilizzato supera una soglia predefinita (esempio 8 KB), `checkSpace.bash` elimina, partendo dal meno recente, un numero di file csv sufficiente per riportare lo spazio utilizzato da tutti i file csv sotto la soglia predefinita.

Aggiungere questa funzionalità al menu testuale visualizzato dallo script `menu.bash`.

Consegna elaborato:

Tutti gli script bash devono essere inclusi in una directory di nome **elaborato_bash**. La directory **elaborato_bash** deve essere compressa in un archivio di nome:

<matricola>_bash.tar.gz

L'archivio **tar.gz** deve essere caricato nell'apposita sezione sul sito di e-learning.