Progetto sistemi operativi
anno 2013-2014

Federico Ponzi Ludovico Ragno

Lista script implementati:
- check.sh
- score.sh
- test.sh
- verifier.sh
- valutazione.sh

Manuale d'uso
- Installazione
- Esecuzione

Lista script implementati:

Premesse generali

Per lo sviluppo di questo programma, è stato scelto un approccio modulare tramite il raggruppamento in funzioni dei processi logici seguiti da ogni script.
Questo vuol dire che il main, che troveremo alla fine di ogni file.sh, sarà composto da richiami a funzioni predichiarate all' interno dello stesso file.
Quando utile, abbiamo scelto di inserire informazioni di **debug** che verranno mostrate se, all' interno del file settings.sh, abbiamo impostato ad 1 la variabile `$DEBUG`.

verifier.sh

Il punto di ingresso del programma di correzione. Non svolge particolari funzioni: esegue la verifica dei parametri e richiama, con ordine, gli script di seguito elencati.

check.sh

Il file check.sh contiene i controlli necessari per verificare la coerenza del file archivio contentente gli elaborati del candidato.
I vari codici di uscita associati agli errori sono stati definiti all' interno del file `./etc/settings.sh`

Una linea di codice interessante che abbiamo utilizzato per verificare che il file `candidato.info` contentesse i corretti dati è stata questa:

```
grep student_name.*$student_name $tmp_dir/candidato.info > /dev/null
|| exit $EARCHIVE_CANDATE_INFO
```

Con questa linea, stiamo cercando usando il comando grep all' interno del file candidato.info, un pattern formato da "`student_name.*$student_name`". Questa **regex** stà a significare che dovrà esserci student_name, seguito da un carattere non alfanumerico, e poi il nome effettivo dello studente fornitoci all' interno di `elaborato_info.sh`. Questo permette flessibilità nella scelta del separatore all' interno di candidato.info.
Usando la redirezione dell' output cancelliamo il risultato di cui ci importa solo sapere se è presente o meno. In caso non sia presente, lo script uscirà con codice di errore `$EARCHIVE_CANDIDATE_INFO`.

test.sh

Lo script `test.sh` è incaricato di impostare l'ambiente per i test da effettuare su ogni esercizio. Una volta effettuate alcune preconfigurazioni, vengono avviati tutti i test Per ogni esercizio provvede poi ad eseguire i test su di esso definiti.

La parte più difficile e interessante è stata scrivere la linea di codice incaricata di avviare l' elaborato del candidato.

```
timeout -k 3 --preserve-status 3 bash -c " cat stdin |
$assessment_tests/elaborato.sh $args" 1> $testdir/stdout 2>
$testdir/stderr
```
Il comando timeout permette di eseguire un comando per un tempo determinato. Dopo di che un segnale di chiusura viene inviato al comando. Per assicurarsi che venga terminato, aggiungiamo il paramentro **-k**. Per mantenere il codice di uscita del comando eseguito (in modo da poterlo salvare all' interno di $testdir/exit) aggiungiamo il parametro **--preserve-status**.
Il comando che eseguiamo è **bash** con parametro **-c** che ci permette di eseguire i comandi che prende come argomento.

score.sh
Questo script viene richiamato dopo **test.sh**, e il suo compito è quello di calcolare il punteggio raggiunto per ogni test effettuato. E' formata da diversi e semplici controlli che confrontano i file attesi con quelli risultanti dall' esecuzione dell' elaborato.
L' unica parte interessante, è come abbiamo utilizzato la funzione di redirezione dell' output con append: `echo $assessmentOut >> $assessment_dir/assessment.out` in modo da poter appendere alla fine del file una riga per ogni test.

valutazione.sh
Similmente a **score.sh**, questo script ha uno svolgimento abbastanza lineare formato da semplici controlli e funzioni matematiche per il calcolo della valutazione dello studente.

Manuale d'uso

Installazione
Per installare lo script bisogna seguire i seguenti passaggi:
1. Estrarre l'archivio contente i file del progetto:
   `root@progettoos:/home/$  tar -zxvf progetto.tar.gz`
2. Aprire il file `etc/settings.sh` ed impostare la variabili evidenziate.
3. Assicurarsi i permessi di esecuzione sul file `verifier.sh`:
   `root@progettoos:/home/$` chmod +x verifier.sh

Esecuzione
Per eseguire il programma, bisogna lanciare il comando:
```
root@progettoos:/home/$ ./verifier.sh Path Tipo Compito Sessione
Matricola
```

Dove:
- `Path` = Path della cartella contenente le **directory** con i **tests** da esegure. Deve essere una path completa.
- `Tipo` = Definisce se è una correzione del **professore** o dello **studente**.

- `Compito` = Definisce se è un **compito** di tipo **shell** o di tipo **system**.
- `Sessione` = Definisce la **data** della sessione
- `Matricola` = Definisce la **matricola** del candidato.

Un esempio di lancio usando i file da noi allegati:

```
./verifier.sh ~/workspace/ self shell 2013_2014_12_01 142323
```

Una volta lanciato il progamma possiamo sapere se il candidato ha **superato** l' esame aprendo il file superamento nella directory a lui assegnata

```
Path/assignments/Tipo/Compito/Sessione/Matricola/superamento
```

Abbiamo allegato un file **basdir.tar.gz**, contenente:

- Gli elaborati forniti dal docente
- Gli elaborati svolti dell' esame di shell di luglio 2014
- I test forniti dal docente
- I test per correggere l' esame di shell di luglio 2014

Assunzione: Le cartelle degli esercizi devono esserci anche se il candidato non ha svolto l' elaborato.