

Script CGI

I programmi **CGI** (*Common Gateway Interface*) sono una modalità per eseguire procedure sul server in risposta alle richieste dell'utente e per restituire i risultati dell'elaborazione sotto forma di pagine Web.

I file CGI sono residenti sul server in una directory convenzionalmente chiamata **cgi-bin**; la directory deve possedere i permessi di esecuzione.

Le applicazioni CGI possono essere scritte usando differenti linguaggi: i più usati sono linguaggio C e *Perl*. Possono essere anche script delle shell di Linux.

Perl (*Practical Extraction and Reporting Language*) è un linguaggio molto simile nella sintassi al linguaggio C, particolarmente adatto per scrivere applicazioni CGI perché possiede molte funzioni per il trattamento di stringhe.

Nella versione *Ubuntu* di Linux, con il server *Web Apache*, i file CGI sono registrati nella directory **/usr/lib/cgi-bin**, che deve avere i permessi 755 e l'utente *root* come *owner*.

Se la directory non esiste, occorre crearla e impostare per essa i permessi, con i seguenti comandi:

```
sudo mkdir /usr/lib/cgi-bin
sudo chmod 755 /usr/lib/cgi-bin
sudo chown root.root /usr/lib/cgi-bin
```

Per rendere disponibili le applicazioni CGI agli utenti che si collegano tramite browser, occorre controllare che la configurazione del server *Web Apache*, modificando il file **000-default.conf**, che si trova nella directory **/etc/apache2/sites-available**.

Nella sezione `<Virtualhost *:80>` del file, prima della linea:

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

occorre inserire la seguente porzione di codice, che imposta la directory dove devono essere collocati i file eseguibili:

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>
```

Dopo aver salvato il file con le modifiche, si deve riavviare il server *Apache*:

```
sudo service apache2 restart
```

La chiamata di una procedura CGI è di solito associata a un pulsante di comando presente all'interno di un form della pagina Web. Il pathname del file CGI deve essere specificato come valore del parametro **ACTION** del tag **FORM** nella pagina in HTML.

Mediante il form, l'utente può anche specificare i valori da passare all'applicazione CGI.

Per comprendere il funzionamento delle applicazioni CGI si considerino i seguenti semplici esempi: essi utilizzano come file CGI uno script della *Bash shell* di Linux, come quelli visti nel volume per la classe 4 della collana *Tecnologie e progettazione di Sistemi Informatici e di Telecomunicazioni*.

Progetto 1

Visualizzazione della data odierna.

L'applicazione è formata da due file:

- la pagina HTML che contiene un titolo e un pulsante di comando (*richiesta.html*);
- il file CGI che contiene uno script della *Bash shell* e che genera in modo dinamico una pagina HTML per visualizzare la data del sistema (*giorno.cgi*).

richiesta.html

```
<html>
<head>
  <title>Richiesta della data</title>
</head>
<body>
  <h1>prova</h1>
  <p>
  <p>
  <form method="post" action="/cgi-bin/giorno.cgi">
    <input type="submit" value="Esegui">
  </form>
</body>
</html>
```

Questo file viene registrato nella directory **/var/www** o in una sua sottodirectory. Il file CGI seguente invece deve essere registrato nella directory **/usr/lib/cgi-bin**.

giorno.cgi

```
#!/bin/bash
# data di oggi
echo "Content-type: text/html"
echo
echo "<html><head>"
echo "<title>prova di script </title>"
echo "</head><body>"
echo "<h1 align='center'>Prova di script </h1>"
echo "<p>"
# attenzione agli apici!
echo "`date`"
echo "<p>"
echo "</body> </html>"
```

Si osservi che la terza e quarta riga dello script precedente sono obbligatorie per il corretto funzionamento del programma:

```
echo "Content-type: text/html"
echo
```

Inoltre il file *cgi* deve essere reso eseguibile attraverso il comando

```
chmod +x giorno.cgi
```

La pagina Web è visualizzata dal browser fornendo l'indirizzo:

```
http://localhost/richiesta.html
```

Quando l'utente preme sul pulsante *Esegui*, il browser lancia l'esecuzione dello script CGI sul server Web, producendo in output la data di oggi all'interno di una pagina HTML generata in modo dinamico.

Il funzionamento corretto dello script CGI può essere preventivamente testato anche fornendo direttamente l'indirizzo al browser:

```
http://localhost/cgi-bin/giorno.cgi
```

Nei casi in cui il browser interrompa l'esecuzione con messaggi di errore, è possibile ottenere informazioni più dettagliate sul tipo di errore consultando il file `/var/log/apache2/error_log`. Il secondo esempio mostra l'esecuzione di uno script CGI che riceve il valore di un parametro fornito dall'utente tramite un form della pagina Web. L'applicazione riprende il problema già svolto nel capitolo 5 come esempio di programmazione nella Bash shell. Qui è mostrato come si possa eseguire la stessa applicazione nel Web.

Progetto 2

Dopo aver creato un file di nome `dizion` contenente per ogni riga un termine italiano e la sua traduzione in inglese, separati dal carattere di tabulazione, si costruisca un'applicazione CGI che accetta il termine fornito dall'utente nel form di una pagina Web e restituisce la sua traduzione in inglese oppure un messaggio di termine non trovato. Controllare anche che l'utente inserisca la parola da tradurre e in caso negativo mandare un messaggio di avvertimento.

L'applicazione è formata da tre file:

- il file `dizion` con i termini italiano e inglese registrato nella directory `/usr/lib/cgi-bin`, per esempio:

```
ciao      hello
giorno    day
night     notte
io        I
tu        you
```

I termini sono separati da una tabulazione.

- la pagina HTML (`inserim.html`), registrata nella directory `var/www` o in una sua sottodirectory, che contiene un titolo, una casella di testo per il termine da cercare, un pulsante di invio e un pulsante di annullamento;
- il file CGI (`ricerca.cgi`), registrato nella directory `/usr/lib/cgi-bin`, che contiene uno script della *Bash shell*: esso genera in modo dinamico una pagina HTML per visualizzare la traduzione del termine o i messaggi di avvertimento per l'utente.

inserim.html

```
<html>
<head>
  <title>Ricerca nel dizionario</title>
```

```

</head>
<body>
  <h1>Cerca un termine nel dizionario</h1>
  <p><p>
  <form action="/cgi-bin/ricerca.cgi" method="get">
    <p>Inserisci qui il termine da cercare:
    <input type="text" name="nome" size=30></p>
    <p><input type="submit" value="Invia">
    <input type="reset" value="Annulla"></p>
  </form>
</body>
</html>

```

Il termine fornito dall'utente nel form della pagina Web è passato allo script CGI nella variabile di ambiente **QUERY_STRING**, che contiene la stringa *nome = termine* dove *nome* indica l'identificativo della casella di testo del form HTML e *termine* è la parola da cercare che l'utente ha scritto nella casella di testo.

ricerca.cgi

```

#!/bin/bash
#ricerca
echo "Content-type: text/html"
echo
echo "<html><head>"
echo "<title>Risultati della ricerca </title>"
echo "</head><body>"
if [ $QUERY_STRING = "nome=" ]
then
  echo "Occorre specificare un termine"
else
  echo "<h1 align='center'>Traduzione</h1>"
  echo "<p>"
  cercato=`echo $QUERY_STRING | cut -c6-30`
  traduzione=`grep $cercato dizion | cut -f2`
  if [ -z $traduzione ]
  then
    echo "il termine non e' presente nel dizionario"
  else
    echo "La traduzione di $cercato = $traduzione"
  fi
fi
echo "<p>"
echo "</body> </html>"

```

Il file CGI deve poi diventare eseguibile con il comando:

```
chmod +x ricerca.cgi
```

La pagina Web è visualizzata dal browser fornendo l'indirizzo

```
http://localhost/inserim.html
```

Quando l'utente preme sul pulsante *Invia*, il browser lancia l'esecuzione dello script CGI sul server Web, producendo in output la traduzione del termine in inglese oppure un messaggio di parola non trovata nel dizionario.