

Laboratorio di sistemi operativi

A.A. 2010/2011

Gruppo 2

Gennaro Oliva

9

Esercizi su script di shell

# Esercizio 19

- 1) Utilizzando awk si scriva un comando che stampi una lista dei file presenti nella directory corrente mostrando solo dimensione e nome
- 2) Si faccia in modo che il comando stampi solo i file maggiori di 1024 byte
- 3) Si calcoli la dimensione occupata in totale dai file regolari con dimensione maggiore di 1024 byte nella directory corrente

# Esercizio 19

- `$ ls -la | awk '{print $5,$9}'`
- `$ ls -la | awk '$5 > 1024 {print $5,$9}'`
- `$ ls -la | grep ^- | awk '\n'$5 > 1024 {TOTAL += $5} ; \nEND {print TOTAL}'`

# Variabili di ambiente

- Visibili a tutti i programmi eseguiti dalla shell
- Forniscono informazioni ai programmi ed eventualmente ne condizionano l'esecuzione
- Il comando `env` ne stampa l'elenco
- Per rendere d'ambiente una variabile dichiarata si usa il comando `export`

```
potter@pocho: ~ (as potter)
potter@pocho:~$ echo $LANG
en_US.UTF-8
potter@pocho:~$ date
Tue Mar 22 11:53:53 CET 2011
potter@pocho:~$ LANG=it_IT.UTF-8
potter@pocho:~$ date
mar 22 mar 2011, 11.54.11, CET
potter@pocho:~$
```

# Le variabili d'ambiente

- Le variabili di ambiente sono accessibili anche agli script di shell
- Quando da una shell interattiva viene invocata l'esecuzione di uno script la shell esegue l'interprete specificato dallo script e gli passa come argomento il file contenente lo script
- Nel caso di script bash quindi, viene eseguita una nuova shell e viene passato come argomento lo script
- La nuova shell ha una copia di tutte le variabili di ambiente della shell interattiva
- Solo le variabili di ambiente vengono copiate

# Modifica delle variabili d'ambiente

- La nuova shell opera sulla propria copia delle variabili e non su quella della shell che ha eseguito lo script
- Se si modifica una variabile d'ambiente all'interno di uno script il suo valore non viene alterato all'interno della shell interattiva

# Esercizio 20

1) Si realizzi uno script `superls` che a seconda del valore della variabile d'ambiente `LIST`: `perm`, `link`, `user`, `group`, `date`; visualizzi il listato dei file nella directory corrente con il campo corrispondente

Esempio:

```
$ export LIST=date
```

```
$ superls
```

```
Apr 7 21:05 .
```

```
Apr 7 21:01 ..
```

```
Mar 31 17:47 00-intro.odp
```

```
Mar 30 09:15 01-unix.odp
```

```
Mar 30 09:05 02-shell.odp
```

```
Mar 30 09:16 03-shell.odp
```

```
Mar 30 09:10 03-shell+vi.odp
```

```
Mar 31 17:09 04-shell_script.odp
```

```
Mar 31 11:10 05-esercizi_shell_script.odp
```

```
Mar 30 09:34 05-vi.odp
```

```
Mar 31 17:13 06-regexpr.odp
```

```
Apr 7 20:37 07-esercitazioni_regexpr.odp
```

```
Apr 7 11:29 08-awk.odp
```

# Esercizio 20

- Al termine dello script precedente si modifichi il valore della variabile LIST e si verifichi che la modifica non altera il valore della variabile della shell interattiva
- Si verifichi che lo script non funziona se la variabile LIST non viene esportata: in una nuova shell si scriva LIST=perm senza eseguire l'export e poi si verifichi che la variabile non è visibile dallo script

# Debug script di shell

- Per effettuare il debug dei nostri script di shell è possibile passare l'argomento `-x` alla bash

```
#!/bin/bash -x
```

- Invocando bash con `-x` vengono visualizzate tutte le linee di comando dopo l'espansione e prima dell'esecuzione
- Le linee di comando vengono precedute dal simbolo `+` per distinguerle dall'eventuale output prodotto

# Esercizio 21

- Si realizzino due script che hanno come interpreti rispettivamente `/bin/echo` e `/bin/cat`, si eseguano e se ne analizzi il comportamento
- Si modifichi lo script precedente in modo da attivare il debug della shell e se ne verifichi il funzionamento

# Utilizzare variabili nei comandi

- Quando si proteggono gli argomenti di sed, awk e grep con gli apici singoli, si preclude l'utilizzo di variabili
- In alcuni casi si possono utilizzare gli apici doppi che non impediscono la sostituzione della variabile con il suo valore
- Esempio:  

```
$ grep “^$USER:” /etc/passwd
```

# Esercizio 22

- Si crei uno script di bash che utilizzando il file listino.txt che trovate sotto la directory /tmp implementi le seguenti funzioni accessibili da un menu
- 1) **Cerca** - Chiede all'utente una stringa da ricercare all'intero del listino ed effettua la ricerca
  - 2) **Aggiungi** - Chiede all'utente il codice del prodotto da aggiungere (primo campo del listino) e la quantità di articoli desiderati, verifica le scelte effettuate e le memorizza in un file carrello
  - 3) **Mostra** - il contenuto del carrello ed il costo totale
  - 4) **Elimina** - un prodotto dal carrello