

ESERCITAZIONE DI GIOVEDÌ 21/03/2024

INIBIZIONI

- 1) Dopo aver verificato che nella directory corrente ci sia un file di nome *p.txt*, con un editor, scrivere il file comandi *prova-comandi.sh* che deve stampare con il comando *echo* il nome assoluto del file *p.txt* e quindi, in modo separato, deve visualizzare la directory corrente: in entrambi i casi inserire frasi che facciano capire all'utente che cosa si sta visualizzando! Verificarne il funzionamento.
- 2) Prendendo ispirazione dal file comandi *provaSost.sh* visto a lezione (scaricabile da GITHUB), con un editor, scrivere un file comandi *prova-inibizioni.sh* che assegna ad una variabile di shell di nome *a* la stringa *ciao*; quindi visualizzare con un unico comando *echo* il valore della variabile *a*, della directory corrente e della espansione del metacarattere ***. Inserire dei commenti significativi. Rendere eseguibile il file comandi e mandarlo in esecuzione. Quale è il risultato?
- 3) Modificare, con un editor, il file comandi *prova-inibizioni.sh* e ricopiare la linea con il comando *echo* andando a inibire TUTTE le sostituzioni. Mandarlo nuovamente in esecuzione. Quale è il risultato?
- 4) Modificare, con un editor, il file comandi *prova-inibizioni.sh* e ricopiare la linea con il comando *echo* andando a inibire SOLO l'ultima sostituzione. Mandarlo nuovamente in esecuzione. Quale è il risultato?

SEMPLICI FILE COMANDI

- 5) Con un editor, scrivere un file comandi di nome *provaExit.sh* che semplicemente contenga il comando *exit* invocato con valore negativo (ad esempio *-1*). Verificarne il funzionamento e giustificare il valore ottenuto!
- 6) Copiare¹ il file comandi *DIR.sh* visto a lezione (scaricabile da GITHUB) nel file comandi di nome *DIRCTL.sh* e quindi, con un editor, aggiungere come prima cosa il controllo che il file comandi sia eseguito esattamente con un solo parametro (CONTROLLO DEI PARAMETRI STRETTO). Verificarne il funzionamento per i vari casi.
- 7) Copiare il file comandi *DIRCTL.sh* dell'esercizio precedente nel file comandi di nome *DIRCTL1.sh* e quindi, con un editor, aggiungere (dopo il controllo sul numero di parametri) il controllo se il parametro è un file o una directory, adattando il codice ai due diversi casi: in particolare, nel caso sia passato il nome di una directory, si dovranno visualizzare le informazioni della directory vista come file, mentre nel caso sia passato il nome di un file il comportamento dovrà essere lo stesso di prima! Verificarne il funzionamento per i vari casi.
- 8) Copiare il file comandi *DIRCTL1.sh* dell'esercizio precedente nel file comandi di nome *DIRCTL2.sh*; aggiungere che, nel caso non si passino parametri (CONTROLLO DEI PARAMETRI LASCO), si devono visualizzare tutte le informazioni dei file e delle sottodirectory della directory corrente, in forma paginata. Osservazione: nel caso nei due esercizi precedenti il controllo sul numero dei parametri è stato fatto con un *if*, in questo caso è più opportuno fare il controllo con un *case*, in modo da intercettare i vari casi: il caso *0* parametri (corretto e in cui bisogna eseguire quanto richiesto e conviene farlo direttamente nel codice dell'alternativa corrispondente del *case*), il caso *1* parametro (corretto e in cui nel seguito, cioè dopo il *case*, verrà eseguito quanto si faceva nel file *DIRCTL1.sh*), i casi *2* o *più parametri*, che come prima dovrà riportare una indicazione di errore (casi NON corretti). Verificarne il funzionamento per i vari casi.
- 9) Con un editor, scrivere un file comandi di nome *un-solo-parametro.sh* che accetti un solo parametro (CONTROLLO DEI PARAMETRI STRETTO) e, in tale caso, riporti sullo standard output se tale parametro rappresenta un nome assoluto, relativo, o relativo semplice. Verificarne il funzionamento per i tre casi.
- 10) Copiare il file comandi *un-solo-parametro.sh* dell'esercizio precedente nel file comandi di nome *un-solo-parametro-bis.sh*; aggiungere SOLO nel caso il nome sia assoluto, il controllo sul 'tipo' di parametro e cioè se è un file o una directory traversabile o nessuno dei due casi. Verificarne il funzionamento per i vari casi.
- 11) Con un editor, scrivere un file comandi di nome *trova-file.sh* che controlli di essere invocato esattamente con 1 parametro (CONTROLLO DEI PARAMETRI STRETTO) e che tale parametro deve essere un nome relativo semplice (come fatto nei due esercizi precedenti). Quindi, dopo la fase di controllo del numero e 'tipo' di parametri, lo script deve verificare se nella directory corrente esiste un file il cui nome è passato come parametro e ne deve stampare il nome assoluto.

¹ Usare esclusivamente il comando *cp* da linea di comando, con eventuali opzioni, se necessario!

- 12) Con un editor, scrivere un file comandi di nome `append.sh` che deve accettare uno o due parametri (CONTROLLO DEI PARAMETRI LASCO), che devono essere nomi di file (in qualunque forma, assoluti/relativi/relativi semplici). Nel caso di un solo parametro, il contenuto dello standard input deve essere scritto in append sul file il cui nome è passato come unico parametro; nel caso di due parametri, il contenuto del file il cui nome è passato come primo parametro deve essere scritto in append sul file il cui nome è passato come secondo parametro. Verificarne il funzionamento per i due casi, sia passando nomi di file esistenti che di file NON esistenti!
- 13) Con un editor, scrivere un file comandi di nome `append1.sh` che si deve comportare in modo simile al file comandi `append.sh`, ma deve prevedere esattamente due parametri (CONTROLLO DEI PARAMETRI STRETTO); devono essere poi introdotti diversi controlli: a) i due parametri devono essere nomi relativi semplici; b) il file il cui nome è passato come primo parametro deve esistere ed essere leggibile; c) il file il cui nome è passato come secondo parametro, se esiste, deve essere scrivibile, altrimenti se non esiste si deve verificare che la directory corrente sia scrivibile. Verificarne il funzionamento nei vari casi.
- 14) Con un editor, scrivere un file comandi di nome `tutti-par.sh` che, dopo aver controllato che il numero di parametri sia maggiore o uguale a 1 (CONTROLLO DEI PARAMETRI LASCO), riporti su standard output tutti i parametri passati sia in forma aggregata che un parametro per volta. Verificarne il funzionamento nei vari casi.
- 15) Copiare il file comandi `tutti-par.sh` dell'esercizio precedente nel file comandi di nome `tutti-par-bis.sh`; aggiungere, all'interno del ciclo in cui si riporta su standard output il valore di ogni parametro passato, l'indicazione del loro 'nome' (\$1, \$2, etc.). SUGGERIMENTO: Usare un contatore e poi stamparne il valore insieme con \\$. Verificarne il funzionamento.
- 16) Copiare il file comandi `LeggiEMostra.sh` visto a lezione (scaricabile da GITHUB) nel file comandi di nome `LeggiEMostr1.sh`; con un editor, modificare la parte di controllo dei parametri in modo da poter funzionare per un numero di parametri maggiore o uguale a 1 (CONTROLLO DEI PARAMETRI LASCO); quindi su ogni parametro passato svolgere le stesse azioni presenti in `LeggiEMostra.sh`, ma usando un case per verificare la risposta dell'utente (come fatto nel file comandi `readCase.sh` visto a lezione e sempre scaricabile da GITHUB). Verificarne il funzionamento nei vari casi.
- 17) Copiare il file comandi `for1.sh` visto a lezione (scaricabile da GITHUB) nel file comandi di nome `for1Bis.sh`; con un editor, modificare il codice in modo che il corpo del ciclo sia eseguito solo per i nomi dei file/directory nascosti. Verificarne il funzionamento, dopo aver verificato di avere almeno un file nascosto.
- 18) Copiare, nuovamente, il file comandi `for1.sh` visto a lezione (scaricabile da GITHUB) nel file comandi di nome `for2.sh`; con un editor, modificare il codice in modo che il corpo del ciclo sia eseguito solo per i nomi che iniziano con la lettera 'p' (nella directory corrente) e, nel caso il nome corrisponda ad un file, si deve anche visualizzarne il contenuto, mentre se è una directory bisogna segnalarlo. Verificarne il funzionamento.
- 19) Copiare il file comandi `for2.sh` dell'esercizio precedente nel file comandi di nome `for2Bis.sh`; aggiungere, nel caso il nome che inizia con la lettera 'p' sia quello di un file, la scrittura del suo nome assoluto in append su file di nome `/tmp/t`. Verificarne il funzionamento e in particolare ricavare il numero di file trovati. Cosa succede se si esegue nuovamente `for2Bis.sh`?
- 20) Copiare il file comandi `for4Bis.sh` visto a lezione (scaricabile da GITHUB) nel file comandi di nome `for4Ter.sh` con l'obiettivo di rendere più generale il codice: infatti, il nome del file usato per l'esecuzione di `for4Bis.sh` deve essere un parametro di invocazione di `for4Ter.sh`. Quindi, si deve prevedere un controllo sul numero dei parametri che deve essere esattamente uguale a 1 (CONTROLLO DEI PARAMETRI STRETTO) e quindi il controllo che tale singolo parametro sia il nome relativo semplice di un file. Sul contenuto del file passato come parametro si deve svolgere il `for` presente in `for4Bis.sh`. Per verificare il funzionamento di questo script, innanzitutto si creino 3 file di testo con nome *pippo.txt* (con 4 linee), *pluto.txt* (con 3 linee) e *paperino.txt* (con 6 linee), poi si crei un file di nome *temp* che deve avere il formato del file *t* usato a lezione. Si verifichi quindi il funzionamento di `for4Ter.sh` passando il file *temp*.
- 21) Si creino poi altri due file con nome *temp1* e *temp2* che devono avere, rispettivamente il seguente formato:
 - *temp1* deve contenere le informazioni del file *temp*, ma riportate su 3 linee in cui in ogni linea ci sia il nome del file e poi il numero di linee di quel file;
 - *temp2* deve contenere le informazioni del file *temp1*, ma riportate su un'unica linea.

Si verifichi usando una shell lanciata con `sh -x` che il risultato dei comandi `echo `cat temp``, `echo `cat temp1`` e `echo `cat temp2`` riporti comunque le informazioni tutte su una singola linea.

Quindi, si verifichi il funzionamento di `for4Ter.sh` passando anche questi due file.

- 22) Con un editor, scrivere un file comandi di nome `primo-par.sh` che, dopo aver controllato che il numero di parametri non sia 0 (CONTROLLO DEI PARAMETRI LASCO), riporti su standard output il primo parametro della linea di comando e anche la lista dei parametri escluso il primo. Verificarne il funzionamento.
- 23) Con un editor, scrivere un file comandi di nome `ultimo-par.sh` che, dopo aver controllato che il numero di parametri non sia 0 (CONTROLLO DEI PARAMETRI LASCO), riporti su standard output l'ultimo parametro della linea di comando e che riporti anche la lista dei parametri escluso l'ultimo. Verificarne il funzionamento.
- 24) Con un editor, scrivere due file comandi; il primo file comandi si deve chiamare `FCP.sh` e deve fare i seguenti controlli: a) essere invocato esattamente con 1 parametro (CONTROLLO DEI PARAMETRI STRETTO); b) tale parametro rappresenti un nome assoluto c) di una directory esistente e traversabile; dopo i controlli, `FCP.sh` deve invocare il secondo file comandi passandogli la directory assoluta. Il secondo file comandi si deve chiamare `file-dir.sh`, dopo essersi spostato nella directory il cui nome è passato come parametro, deve scorrere l'intero contenuto della directory e deve riportare la stringa `[F]` prima del nome assoluto di ogni file regolare ed una stringa `[D]` prima del nome assoluto di ogni directory; prima di terminare, da ultimo, deve riportare su standard output il numero totale di nomi di file stampati e il numero totale di nomi di directory stampate. Verificarne il funzionamento. **NOTA BENE: il file comandi `file-dir.sh` NON deve essere ricorsivo.**

SEMPLICI FILE COMANDI RICORSIVI

- 25) Copiare il file comandi `BeginC.sh` visto a lezione (scaricabile da GITHUB; N.B. si scarichi anche il file comandi ricorsivo `Cercafile.sh`) nel file comandi di nome `BeginC-sbagliato.sh` dove si vada a cancellare o a commentare le due righe corrispondenti al settaggio e all'esportazione della variabile di ambiente `PATH`. Verificarne il funzionamento per i vari casi di invocazioni errate e poi di invocazioni giuste (con due e un parametro) viste a lezione: in questi ultimi due casi, si ha funzionamento corretto?
- 26) Prendendo ispirazione dai file comandi `BeginC.sh` e `Cercafile.sh` visti a lezione (scaricabili da GITHUB), con un editor, scrivere due file comandi `BeginC-dir.sh` e `Cercadir.sh`. Il primo file comandi si deve comportare come `BeginC.sh` con la sola differenza che il nome relativo passato come primo o secondo parametro dovrà poi essere considerato come nome relativo semplice di una directory (e non di un file). Quindi, il secondo file comandi (che deve essere ricorsivo) deve cercare tutte le directory con il nome specificato (e non i file). **SI FACCIA PARTICOLARE ATTENZIONE CHE NELLA RICERCA DELLE DIRECTORY CON IL NOME RELATIVO PASSATO SI DEVE CONSIDERARE ANCHE LA RADICE DELLA GERARCHIA SU CUI SI EFFETTUA LA RICERCA!** Verificarne il funzionamento per i vari casi. **Suggerimento:** per cercare il nome della directory il cui nome relativo è `$2` usare un case sul nome assoluto della directory `$1`!

LN

- 27) Creare una sottodirectory dandogli nome `provaDir` e creare in tale sottodirectory un link `HARDWARE` di nome sia `111` che riferisca un file presente nella propria HOME directory; e verificare sia il numero di link che l'i-number sia del link che del nome 'originale'; modificare il file utilizzando il link e verificare la modifica usando il nome 'originale' del file.
- 28) Creare una copia, sempre nella sottodirectory precedentemente creata, di un file presente nella propria HOME directory e verificare sia il numero di link che l'i-number del file destinazione e del file 'sorgente'.
- 29) Modificare il file 'copia' e verificare che il file originale non sia stato modificato!
- 30) Cancellare il file presente nella propria HOME directory che è stato usato per creare il un link `HARDWARE 111` e quindi verificare sia il numero di link che l'i-number di tale link oltre che verificarne il contenuto.
- 31) Rinominare un file della propria HOME directory.
- 32) Spostare uno dei file della propria HOME directory sempre nella sottodirectory precedentemente creata, mantenendo lo stesso nome.
- 33) Spostare uno dei file della propria HOME directory sempre nella sottodirectory precedentemente creata, cambiandogli nome.