

Esercizio 1

- Creare una directory eser2
- Creare, all'interno di eser2, un insieme di file e di sottodirectory
- Scrivere in eser2 uno script shell che
 - Crei la directory "MovedFiles"
 - Rinomini tutti i file “.txt” in “.xtx”
 - Sposti tutti i file (ad eccezione delle sottodirectory) all'interno di "MovedFiles"

Esercizio 2

Scrivere uno script shell che crei un nuovo file (per es. "FileList") nella directory corrente contenente la lista dei file preponendo

- al nome di un file “regolare” la lettera 'F'
- al nome di directory la lettera 'D'

Lo script ordina il file così creato e lo visualizza su standard error.

Esercizio 3

Scrivere uno script shell che riceve in input due argomenti:

- primo argomento: un nome di directory
- secondo argomento: una stringa,

Lo script:

- Visualizza un messaggio d'errore e termina se:
 - il numero di parametri passati dall'utente e' diverso da due
 - Il primo argomento non corrisponde ad una directory.
- In caso non vi siano errori, visualizza:
 - i nomi dei file all'interno della directory specificata, che contengano la stringa.

Esercizio 4

Scrivere uno script shell che prenda in input tre parametri:

- Due valori numerici
- Un carattere che denota un operatore: s=somma, m=sottrazione, d=divisione e p=prodotto.

Lo script esegue l'operazione indicata e visualizza il risultato.

Lo script deve controllare che:

- il numero di parametri sia corretto. In caso contrario visualizza un errore.
- il terzo parametro appartenga all'insieme dei parametri consentiti (e in caso contrario visualizzi un errore)

Es.:

- Input: esercizio4 4 5 s
- Output: 9

Esercizio 5

Si scriva (e si esegua) uno script che riceve due parametri n ed m .

- Per ogni $i=1,\dots,n$
- Se la directory sorgenti_ i non esiste
 - Crea la directory sorgenti_ i , e visualizza “Directory sorgenti_ i creata con successo” od un messaggio di errore
 - Crea la directory header_ i , e visualizza “Directory header_ i creata con successo” od un messaggio di errore
 - Crea in sorgenti_ i , I file source_ i _ j .c, per $j=1,\dots,m$
 - Crea in header_ i .h, I file header_ i _ j .c, per $j=1,\dots,m$
- Riscrivere lo stesso script utilizzando I costrutti for, while ed until.

Esercizio 6

Scrivere uno script shell che, per ogni utente connesso visualizzi:

- il numero di processi attivi sulla macchina;
- Il PID minimo
- Il PID massimo

Esercizio 7

Scrivere uno script shell che prende come parametro il nome di una directory D ed elenchi, nell'ordine:

- Tutti i file regolari contenuti nell'albero radicato in D
- Tutti i file eseguibili contenuti nell'albero radicato in D
- Tutte le directory contenute nell'albero radicato in D.

Esercizio 8

Scrivere uno script “driver” per gli esercizi appena visti.

1. Lo script visualizza un semplice menu testuale.
 - Ad es. 1) esercizio1, 2) esercizio2... 9) fine
2. Prende in input un valore ed esegue lo script corrispondente
 - Se lo script necessita di parametri, questi vengono richiesti all'utente.
3. Lo script termina se l'utente seleziona l'opzione di terminazione (nell'esempio il numero 9) altrimenti torna al passo 1.

Utilizzare le funzioni per implementare le differenti opzioni.