UNIVERSITÀ DEGLI STUDI DI PARMA
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica
a.a. 2007-2008

CORSO DI SISTEMI OPERATIVI A - ESERCITAZIONE 1

1 Accesso al Sistema

Questa prima esercitazione ha lo scopo di fornire le indicazioni essenziali per familiarizzare con il sistema operativo Linux.

Come attivare una sessione :

1. se il PC è spento o sta eseguendo Windows, (ri)avviarlo, selezionare Linux e premere il tasto Enter ;
2. attendere il completamento della fase di boot del sistema operativo ;
3. alla richiesta di *Username* e *Password*, inserire i dati del proprio account ;
4. se il login non ha successo, riprovare almeno una volta e in caso di nuovo insuccesso verificare con i tecnici i dati del proprio account.

Al termine della sessione di lavoro, selezionare "Termina sessione → Disconnetti" dal menu grafico e attendere la disconnessione.

2 Uso della Shell

Per potere dare comandi al sistema operativo occorre accedere ad una riga di comando da cui poterli digitare, ovvero aprire una shell. Solitamente questo è possibile cliccando sull'icona in basso a sinistra raffigurante un monitor con davanti un guscio (shell) di conchiglia. Se riscontrate problemi contattate un assistente.

Esistono vari tipi di shell (bash, tcsh,...), che presentano poche ma significative differenze. Di norma le impostazioni del vostro account vi faranno accedere ad una shell di tipo bash.

Chi sono? Ovvero, qual è il nome dell'utente proprietario di questa shell? Utilizzare il comando *whoami*.

NOTA 1 FONDAMENTALE: ogni comando di sistema ha il suo manuale di riferimento, che può essere consultato con il comando ***man*** *<nomecomando>* (ad es. *man ls*). Scorrere il contenuto del manuale utilizzando le freccette della tastiera. Per uscire da un manuale premere "q". Possono esistere più versioni (pagine) di manuale per lo stesso comando (questo perché ad esempio un comando o una funzione può esistere sia per la shell sia per il linguaggio C): per visualizzarle tutte aggiungere l'opzione -a (ad es. *man -a printf*, con "q" seguito da invio si passa al manuale successivo). Per selezionare una pagina precisa di manuale, indicarne il numero, es. *man 3 printf*.

NOTA 2: ogni volta che si richiede ad uno shell l'esecuzione di un comando, viene ricercato un file eseguibile con lo stesso nome del comando all'interno dei direttori specificati nella variabile d'ambiente PATH. Per visualizzare il percorso assoluto del file eseguibile individuato, in una shell di tipo bash utilizzare il comando *type <nomecomando>* (ad es. *type*

cp). Al riguardo è utile anche il comando *whereis* (es. *whereis ls*). Nelle shell di tipo tcsh il comando da usare al posto di *type* è *which*.

NOTA 3: nelle tastiere italiane non sono presenti alcuni caratteri essenziali come ~ (tilde, viene utilizzata per indicare la home dell'utente), { (parentesi graffa aperta) e } (parentesi graffa chiusa). Per digitarli, utilizzare le seguenti combinazioni di tasti:

- Alt-Gr+ì → ~
- Alt-Gr+7 → {
- Alt-Gr+0 → }

NOTA 4: durante la digitazione dei comandi, abituarsi ad utilizzare il tasto Tab, che aiuta a completare in modo corretto e più veloce i comandi (vedi paragrafo 5.1).

NOTA 5: per interrompere l'esecuzione di un comando, usare la combinazione di tasti Ctrl^c (tenere premuto il tasto "Ctrl" e premere "c").

3 Gestione del File System

3.1 File e Direttori

Esercizio guidato:

1. Esistono vari modi per creare un file. Il modo più semplice per creare un file vuoto è quello di usare il comando *touch*: *touch pippo*
   Suggerimento: scrivere "tou" e poi premere il tasto Tab per vedere completato automaticamente il comando.
   Dopo aver usato il comando *touch* per creare il file pippo non si ottiene alcuna conferma dell'avvenuta esecuzione dell'operazione. Questo atteggiamento è tipico dei sistemi Unix i cui comandi tendono a non manifestare il successo delle operazioni eseguite. Si può comunque verificare: *ls -l pippo*
2. rimuovere il file *pippo*: *rm pippo*
3. creare il direttorio *prova*: *mkdir prova*
4. copiare il file *pippo.txt* da */home/soa/eserc1* (direttorio condiviso accessibile in sola lettura) a *prova*: *cp /home/soa/eserc1/pippo.txt prova*
5. portarsi nel direttorio *prova*: *cd prova*
6. verificare la posizione del direttorio corrente nel File System: *pwd* (Print Working Directory)
7. visualizzare il contenuto del direttorio corrente (*prova*): *ls*
8. identificare il tipo del file *pippo.txt*: *file pippo.txt*
9. visualizzare il contenuto del file *pippo.txt*: *cat pippo.txt*
10. tornare nel direttorio a monte: *cd ..*
11. (Il punto che si trova in fondo al prossimo comando fa parte del comando stesso, e serve ad indicare la directory corrente) Spostare il file *~/prova/pippo.txt* nel direttorio corrente: *mv ~/prova/pippo.txt .*

12. rimuovere il direttorio *prova*: *rmdir prova*

13. portarsi nel direttorio */usr/local*: *cd /usr/local*
    portarsi in */usr/bin* effettuando uno spostamento relativo: *cd ../bin*
    verificare la posizione del direttorio corrente nel File System: *pwd*

14. portarsi in */var/tmp* effettuando uno spostamento relativo cumulato: *cd ../../var/tmp*
    verificare la posizione del direttorio corrente nel File System: *pwd*

15. tornare nel proprio direttorio principale: *cd ∼*

16. creare una struttura articolata di direttori:
    mkdir carbonio
    mkdir carbonio/idrogeno
    mkdir carbonio/idrogeno/elio

17. provare a rimuovere il direttorio *carbonio*: *rmdir carbonio*
    il comando precedente non funziona perchè *carbonio* contiene dei direttori; per rimuoverlo occorre utilizzare la ricorsione: *rmdir -p carbonio/idrogeno/elio*
    NOTA: per ottenere lo stesso risultato si può utilizzare *rm -r carbonio*, ma *rmdir -p* permette di cancellare singoli rami di un albero di direttori.

3.2 Linking

Esercizio guidato:

1. creare un link simbolico al direttorio */home/soa/eserc1*: *ln -s /home/soa/eserc1 es1*

2. verificare che il link è stato creato correttamente: *ls -l*

3. creare un hard link al file *es1/pluto.txt*: *ln es1/pluto.txt pluto2.txt*

4. verificare la condivisione dell'i-node tra il file originale e l'hard-link, eseguendo *ls -l* (si osservi il numero nella seconda colonna)

5. visualizzare il contenuto del file *pluto2.txt* (l'hard link): *cat pluto2.txt*

6. cancellare *pluto2.txt*, rispondendo *y* (yes) alla richiesta di autorizzazione.

3.3 Protezione

Esercizio guidato:

1. copiare il file */home/soa/eserc1/pippo.txt* nel proprio direttorio principale

2. visualizzare il contenuto dettagliato del direttorio corrente: *ls -l*

3. cambiare i diritti di *pippo.txt* in modo che sia tolto il permesso in lettura a tutti gli utenti, anche al proprietario: *chmod a-r pippo.txt*

4. visualizzare nuovamente il contenuto dettagliato del direttorio (cosa è cambiato?)

5. verificare che il file non è leggibile: *cat pippo.txt*

6. ripristinare il permesso in lettura e togliere quello in scrittura: *chmod a+r-w pippo.txt*

7. verificare che il file non è scrivibile: *cp /home/soa/eserc1/pluto.txt pippo.txt*
   Un tentativo di sovrascrittura genera una segnalazione di errore, come nell'esempio appena visto, così come qualunque altro tentativo di modificare il suo contenuto.

8. aggiungere il permesso di scrittura sul file *pippo.txt* al solo proprietario: *chmod u+w pippo.txt*
   NOTA: oltre all'opzione *u* (utente proprietario) ed *a* (tutti gli utenti, equivalente a *ugo*), ci sono anche *g* (permessi al gruppo) e *o* (permessi a tutti gli utenti esterni al gruppo);

9. Il comando *chmod* può operare anche sui direttori; creare una cartella *prova* nel proprio direttorio principale e poi togliere il permesso in esecuzione a *prova* con il comando: *chmod a-x prova*. Provare ora ad accedere alla directory: *cd prova*

Esercizio proposto:
ripristinare i diritti di esecuzione del direttorio *prova* solo per il proprietario.

4 Altri comandi di sistema

4.1 Comandi di stato

Provare i comandi visti a lezione: *date*, *time <nomecomando>*, *who*, *ps [-elf]*, *top*, *free*, *du*.

4.2 Ricerca

Provare l'esempio visto a lezione relativo al comando *grep*.

Provare anche il seguente comando: *find / -name bash*

Questo comando esegue una ricerca per i file e le directory denominati *bash* all'interno di tutte le directory che si articolano a partire dalla directory radice (/). Il file viene trovato, ma tutte le volte che *find* tenta di attraversare directory per cui non si ha il permesso, si ottiene una segnalazione di errore.

5 Uso dello Shell

5.1 Completamento automatico

Il completamento automatico è un modo attraverso cui lo Shell aiuta l'utente a completare un comando. La richiesta di completamento viene fatta attraverso l'uso del tasto [Tab].
Si preparano alcuni file di esempio (i nomi utilizzati sono volutamente lunghi):

touch microcontrollore
touch microfono
touch ballatoio

Supponendo di voler utilizzare questi nomi all'interno di una riga di comando, si può essere un po' infastiditi dalla loro lunghezza. Utilizzando il completamento automatico si risolve il problema:

ls bal[Tab]

Dopo avere scritto solo *bal*, premendo il tasto [Tab] si ottiene il completamento del nome, dal momento che non esiste alcun file o direttorio (nella posizione corrente) che inizi nello stesso modo.
Il completamento automatico dei nomi potrebbe essere impossibile. Infatti, potrebbe non esistere alcun nome che coincida con la parte iniziale già inserita, oppure potrebbero esistere più nomi composti con lo stesso prefisso. In quest'ultimo caso, il completamento si ferma al punto in cui i nomi iniziano a distinguersi:

ls mic[Tab]ro

In questo caso, il completamento si spinge fino a *micro* che è la parte comune dei nomi microcontrollore e microscopio. Premendo nuovamente [Tab] vengono visualizzati i possibili completamenti rimasti. Per poter proseguire occorre aggiungere un'indicazione che permetta di distinguere tra i due nomi. Volendo selezionare il primo di questi nomi, basta aggiungere la lettera *c* e premere nuovamente il tasto [Tab]:

ls mic[Tab]roc[Tab]ontrollore

5.2 I metacaratteri

L'utilizzo di metacaratteri rappresenta una forma alternativa di completamento dei nomi. Infatti è compito dello Shell trasformare i simboli utilizzati per questo scopo.
Per questo esercizio si utilizzano i file creati nella sezione precedente: microcontrollore, microscopio e ballatoio.
L'asterisco rappresenta una sequenza indefinita di caratteri di qualunque tipo, esclusa la barra di separazione tra le directory. Per cui l'asterisco utilizzato da solo rappresenta tutti i nomi di file disponibili nella directory corrente.

*ls micro**

Questo comando fa in modo che lo Shell elenchi tutti i file il cui nome inizia per *micro*.
Il punto interrogativo rappresenta esattamente un carattere qualsiasi. Con il comando:

ls ??????o?o

si ottiene il seguente output:

ballatoio microfono

Le parentesi quadre vengono utilizzate per delimitare un elenco o un intervallo di caratteri. Rappresentano un solo carattere tra quelli contenuti, o tra quelli appartenenti all'intervallo indicato.

Osservare cosa fa il comando: *ls ?????[abdf]**
Osservare anche cosa fa: *ls ?????[c-f]**

Il fatto che lo Shell sostituisca alcuni caratteri impedisce di fatto il loro utilizzo nei nomi

di file e direttori. Se esiste la necessità, è possibile evitare la sostituzione di questi facendoli precedere dal carattere \, che funge da carattere di "escape". Ad esempio, per creare il file (vuoto) *sei∗otto*, si usa:

touch sei\∗otto

Esercizio proposto:
creare un nuovo direttorio e copiare al suo interno tutti i file di */home/soa/eserc1* che contengono nel nome la stringa *pippo*.

5.3 Redirezione dell'I/O

Esercizio guidato:

1. scrivere in un file il contenuto di */home/soa/eserc1*, ordinato in base alla data di creazione: *ls -t /home/soa/eserc1 > paperino.txt*
2. visualizzare il contenuto del file generato: *cat paperino.txt*
3. è possibile redirigere l'input e l'output contemporaneamente; come prova, scrivere in un nuovo file il contenuto, questa volta ordinato alfabeticamente, del file creato precedentemente: *sort < paperino.txt > topolino.txt*

5.4 Piping dei comandi

Visualizzare il numero dei file contenuti nel direttorio *bin*: *ls /bin | wc -l*

Esercizio proposto:
utilizzando il piping dei comandi e la redirezione dell'output, scrivere in un file di testo i 10 file più recenti del direttorio */home/soa/eserc1* (suggerimenti: utilizzare il comando *head* e leggere il manuale del comando *ls*, con *man ls*).

5.5 Modalità di esecuzione

In tutti gli esercizi svolti finora i comandi sono stati eseguiti in modalità *foreground*. Come visto a lezione esiste una seconda modalità di esecuzione detta *background*, che si ottiene ponendo alla fine della linea di comando il simbolo &. In quest'ultima modalità lo shell padre non attende il completamento dell'esecuzione del comando.
Esercizio guidato:

1. digitare il seguente comando: *ls -lR > prova.txt &*
2. lo shell è immediatamente disponibile, quindi digitare: *date*

5.6 Controllo della modalità di esecuzione dei comandi

Il comando *jobs* fornisce l'elenco dei comandi attualmente eseguiti in background, indicandone il job number (tra parentesi quadre), lo stato e il nome.
Esercizio guidato:

1. eseguire kwrite (programma per l'editing di testi) in background: *kwrite &*
2. eseguire mozilla (web browser) in background: *mozilla &*

3. eseguire il comando: *jobs*

E' possibile portare da background a foreground l'esecuzione di un comando utilizzando il comando *fg*. Se si sono più esecuzioni in background, bisogna specificare quale portare in forground indicandone il job number.
E' altresì possibile effettuare l'operazione inversa, cioè portare un'esecuzione da foreground a background; per fare ciò bisogna anzitutto sospendere l'esecuzione del comando con la combinazione di tasti [Ctrl^z], e poi utilizzare il comando *bg*.
Esercizio guidato:

1. portare in foreground l'esecuzione di mozilla: *fg %<numero di job di mozilla>*
2. sospendere l'esecuzione di mozilla: [Ctrl^z]
3. portare in background l'esecuzione di mozilla: *bg*