Shell BASH

Adattato da slides del prof. Alberto Finzi:
http://wpage.unina.it/alberto.finzi/didattica/LSO/materialeLSO2021.html

Sommario: comandi concatenabili

Solo a **inizio** pipe: `echo`, `ls`, etc.
(tutti quelli che scrivono su stdout)

Anche al **centro**: `wc`, `sort`, `uniq`, `grep`, `cat`, `head`, `tail`

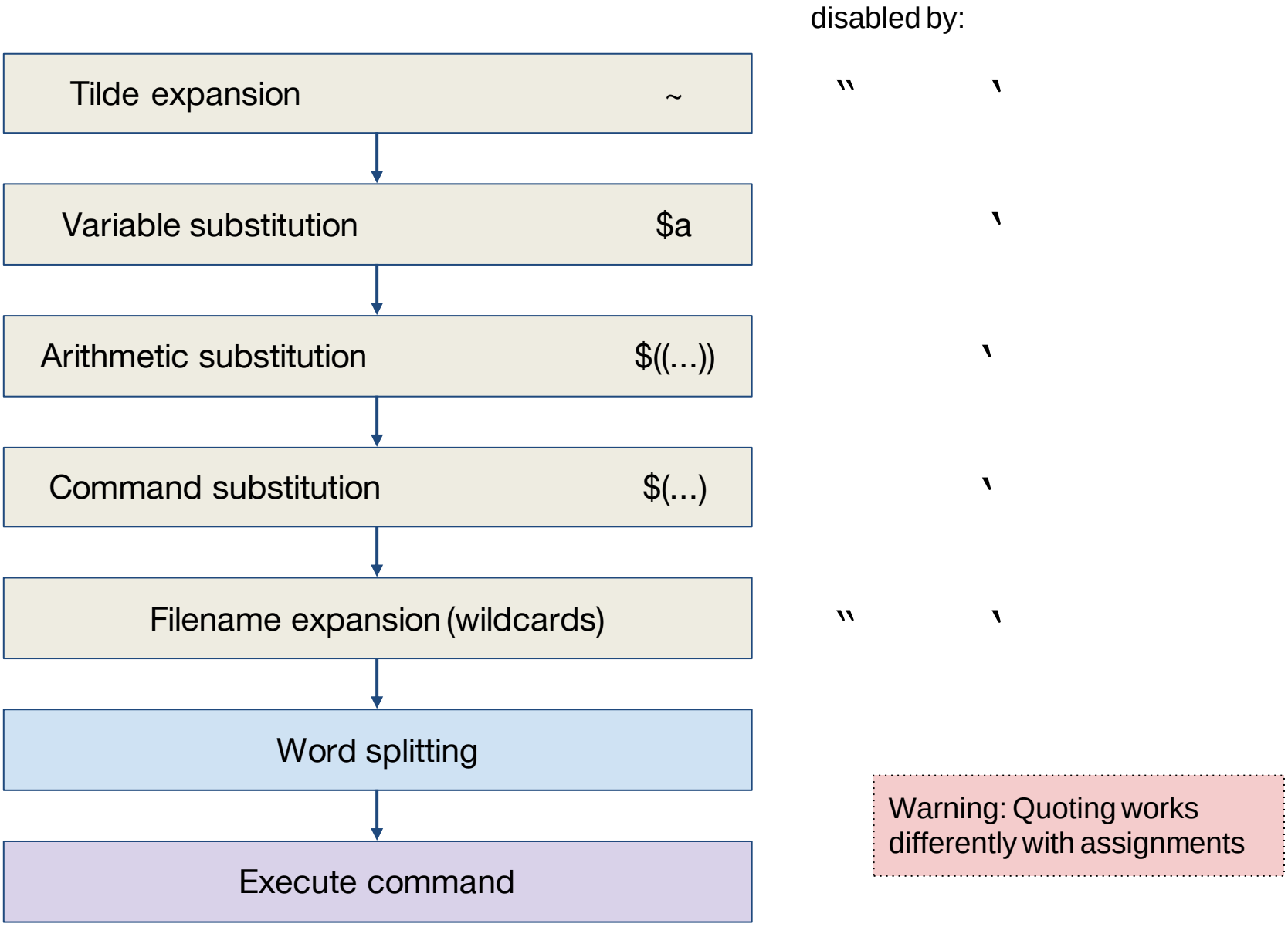
- se richiamati senza argomenti, leggono da stdin
- scrivono su stdout

Solo a **fine** pipe: `less` (paginatore interattivo)

Esercizio n° 0

- 0a) Creare una cartella **EsercitazioneLSO-1** nella directory di lavoro
- 0b) Creare un file testo chiamato **provaFile.txt** che contiene username e hostname
- 0c) Creare una variabile che contiene il contenuto di provaFile
- 0d) Aggiungere alla variabile il nome del file
- 0e) Creare un file che contiene il contenuto della variabile

Shell expansions and substitutions



Esercizi

1. Creare un file che si chiami come l'utente corrente
2. Assegnare alla variabile *x* l'elenco dei file che cominciano con un punto
3. Scrivere alcune parole nel file *nomi.txt*. Successivamente, per ogni parola contenuta nel file, creare un file con nome uguale a quella parola

wc (word count)

wc [*options*] [*file...*]

fornisce il numero dei codici di interruzione di riga (in pratica il numero delle righe), delle parole o dei caratteri contenuti in ***file.*** Senza opzioni fornisce, nell'ordine suddetto, ciascuna delle precedenti informazioni.

Alcune opzioni:

-c	emette solo il numero complessivo di caratteri di ***file***.
-w	emette solo il numero complessivo di parole in ***file***.
-l	emette solo il numero di righe in ***file***.

Esempi di esecuzione

```
gio$ wc which_manpage
132       239       2083 which_manpage
gio$ wc -c which_manpage
2083 which_manpage
gio$
```

Esercizi

1. Assegnare alla variabile *x* il numero di righe di un file a vostra scelta
2. Contare i file della directory corrente che contengono una “z” nel nome
3. Contare i file nella directory corrente che non contengono una “z” all’inizio del nome

Sort

sort [*options*] [*file...*]

permette di (ri)ordinare o fondere insieme il contenuto dei file passati come parametri, oppure di (ri)ordinare le linee passategli in input.

In assenza di opzioni che definiscano diversi criteri di ordinamento, quest'ultimo avviene in base al primo campo ed è alfabetico.

Alcune opzioni:

-f	ignora le differenze tra lettere minuscole e maiuscole
-n	considera numerica anzichè testuale la chiave di ordinamento
-r	ordina in senso decrescente anzichè crescente
-o *fileout*	invia l'output a fileout anzichè sull'output standard
-t *s*	usa s come separatore di campo
-k *s1,s2*	usa i campi da s1 a s2-1 come chiavi di ordinamento

Esercizi

1. Elencare i file della directory corrente in ordine alfabetico *inverso*
2. Scrivere nel file “elenco” l'elenco dei file nella directory corrente, in ordine alfabetico

head & tail

Comando/Sintassi	Descrizione
head [*-numero*] file	visualizza le prime 10 (o *-numero*) linee di un file
tail [*-numero*] file	visualizza le ultime 10 (o *-numero*) linee di un file

Esempio d'uso head:

head -40 filename
oppure
head -n 40 filename

Esempio d'uso tail:

tail -30 filename

Esercizio

- Scrivere una combinazione di comandi Unix che consenta di visualizzare:
  1. la **terza e la quarta** riga del file **provaFile.txt**
  2. le **penultime** 3 righe del file **provaFile.txt**
  3. l' **n-esima** riga del file **provaFile.txt**

Soluzione 1

head –4 provaFile1.txt | tail -2

Soluzione 2

tail –4 provaFile1.txt | head -3

Soluzione 3

head -n provaFile1.txt | tail -1

Word splitting

L'ultima fase prima di eseguire un comando consiste nella **suddivisione in parole**

La variabile IFS (internal field separator) definisce i separatori

Di default, IFS="<space><tab><newline>"

Come effetto collaterale, il word splitting sostituisce i newline con spazi

In una directory con molti file, confrontare l'output di "ls" con quello di "echo $(ls)"

Riferimenti

Bash Guide for Beginners (online)