Riusciamo a scrivere di meno?

- Vogliamo eseguire i nostri compiti nel modo più veloce possibile, quindi è naturale cercare di scrivere il meno possibile.
- Possiamo utilizzare dei caratteri speciali, come *** e *?* per abbreviare il comando
- Il carattere speciale *** vuol dire: sostituisci con zero o più caratteri qualunque. Esempi:
  - *qualcosa** corrisponde a tutte le stringhe che hanno *qualcosa* come prefisso
  - **qualcosa* corrisponde a tutte le stringhe che hanno *qualcosa* come suffisso
  - **qualcosa** corrisponde a tutte le stringhe che hanno al loro interno *qualcosa* (compresa la stringa *qualcosa*)
- Il carattere speciale *?* vuol dire: sostituisci un carattere qualunque. Esempi:
  - *qualcosa?* corrisponde a tutte le stringhe che hanno *qualcosa* come prefisso e che hanno un ulteriore carattere (es. qualcosaa, qualcosab, qualcosa1, qualcosa#, …)
  - *?qualcosa* corrisponde a tutte le stringhe che hanno *qualcosa* come suffisso, preceduto da un carattere qualunque (es. aqualcosa, 9qualcosa, …)
  - *?qualcosa?* corrisponde a tutte le stringhe che hanno al loro interno la sottostringa *qualcosa,* preceduto e seguito da un carattere
- Sia *** che *?* possono anche comparire in posizioni che non siano alle estremità della stringa. Esempio: *qual*cosa*, *qual?cosa*

Scriviamo di meno

- Come posso utilizzare i caratteri che abbiamo visto precedentemente per scrivere in modo piu' conciso: *cat test_1.txt test_2.txt test_3.txt* ?
  - *cat test_?.txt*
  - *cat test_*.txt*
  - *cat test*.txt*
  - *cat test**
  - *cat t**

Creiamo un nuovo file che contenga il contenuto dei vari test

- Come possiamo creare un nuovo file con all'interno il contenuto dei vari file di test che abbiamo scritto prima?
  - *cat t* > combined.txt*
- Cosa succede se ripetiamo questo stesso comando?

- E se invece volessimo effettivamente "appendere" un testo ad un file esistente?
  - Utilizziamo l'operatore *>>*
- Proviamo:
  - *cat t* >> combined.txt*
  - *echo "ho appeso una linea" >> combined.txt*
  - *cat combined.txt*
- Ripetiamo i comandi un po' di volte (provate ad utilizzare la freccia in su), fino ad ottenere un file più lungo della dimensione dello schermo. Come possiamo leggere tale file?
  - *less combined.txt*

Spostare file

- E' possibile spostare un file o una cartella utilizzando il comando *mv* (**m**o**v**e)
  - Il comando *mv combined.txt dir1* sposta il file *combined.txt* all'interno della cartella *dir1*
- Verifichiamo se il file è stato effettivamente spostato

```
corso@corso-os:~$ mv  combined.txt dir1
corso@corso-os:~$ ls dir1
combined.txt
corso@corso-os:~$
```

- Ora supponiamo di aver cambiato idea e di non volere il file in *dir1*, ma nella directory dove si trovava inizialmente. Come facciamo? Diverse strade, proviamone alcune:
  - Posso specificare come percorsi di partenza e destinazione dei percorsi assoluti
  - Posso spostarmi nella cartella che attualmente contiene combined.txt ed utilizzare dei percorsi relativi per spostare il file una directory sopra
  - Posso rimanere dove mi trovo ed utilizzare i percorsi relativi per spostare il file nella directory corrente
  - ...

Spostamenti multipli di file e cartelle

- *mv* consente di spostare con un solo comando diversi elementi in una cartella di destinazione
  - Esempio: *mv combined.txt test_* dir3 dir2* sposta all'interno di *dir2* gli elementi *combined.txt, test_*, dir3*
  - Notare che gli elementi oggetto dello spostamento possono essere sia file che cartelle, mentre l'elemento di destinazione deve necessariamente essere una cartella (non avrebbe senso altrimenti)

```
corso@corso-os:~$ mv combined.txt test_* dir3 dir2
corso@corso-os:~$ ls dir2
combined.txt  dir3  test_1.txt  test_2.txt  test_3.txt
corso@corso-os:~$
```

- Esercizio: spostare ancora una volta il file *combined.txt* da *dir2* a *dir4/dir5/dir6*

Copiare e rinominare

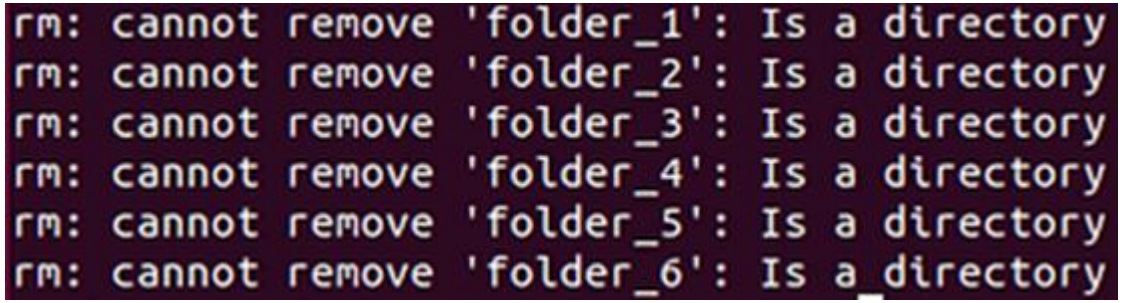
- Il comando per la copia di un file è molto simile a quello per lo spostamento:
  - *cp <percorso di origine> <percorso di destinazione>*
- Come copiamo il file combined.txt che sta dentro dir4/dir5/dir6 all'interno della directory corrente?
  - *cp dir4/dir5/dir6/combined.txt .*
- A questo punto, supponiamo di voler creare una copia di backup di *combined.txt* all'interno della stessa cartella, chiamata *backup_combined.txt.* Come facciamo?
  - *cp combined.txt backup_combined.txt*

- E se il nome che abbiamo scelto non dovesse piacerci? Supponiamo di voler rinominare il file *backup_combined.txt* in *combined_backup.txt.*
  - *mv backup_combined.txt combined_backup.txt*

- Esercizio: rinominare tutte le cartelle con lo spazio che avevamo creato prima.

Cancellare file e cartelle

- Attenzione! La cancellazione di un file o di una cartella da shell non utilizza il meccanismo del cestino
  - Questo vuol dire che un elemento, una volta cancellato, è perso.
- La cancellazione di un file avviene tramite il comando *rm* (**r**e**m**ove), seguito da una lista di file da rimuovere. Esempio:
  - *rm dir4/dir5/dir6/combined.txt combined_backup.txt*
- Proviamo ora a rimuovere anche le cartelle *folder_** che avevamo creato precedentemente.



Cancellare cartelle

- Per la cancellazione di cartelle dobbiamo utilizzare il comando *rmdir*.
  - Proviamo di nuovo ad eliminare le cartelle *folder_ **

```
rmdir: failed to remove 'folder_6': Directory not empty
```

- *rmdir* richiede che la cartella da rimuovere sia vuota. Non deve contenere ne' sottocartelle ne' files.

- Esiste un modo per rimuovere una cartella e tutto il suo contenuto, incluse le sue sottocartelle e, a loro volta, le loro sottocartelle, e così via
  - Si utilizza l'opzione *–r* di *rm* per indicare una rimozione ricorsiva.

Altri comandi utili: wc

- *wc* (**w**ord **c**ount) consente di contare il numero di parole all'interno di un file
  - Dal manuale di *wc:* "A word is a non-zero-length sequence of characters delimited by white space"
- Non solo – *wc* consente di contare anche:
  - Il numero di byte in un file (opzione *–c*)
  - Il numero di line in un file (opzione *–l*)
- Proviamo ad utilizzarlo per contare il numero di linee in *combined.txt …*

```
corso@corso-os:~$ wc -l combined.txt
54 combined.txt
corso@corso-os:~$
```

- … Ed il numero di caratteri…

```
corso@corso-os:~$ wc -c combined.txt
1044   combined.txt
corso@corso-os:~$
```

- … e per finire, senza argomenti

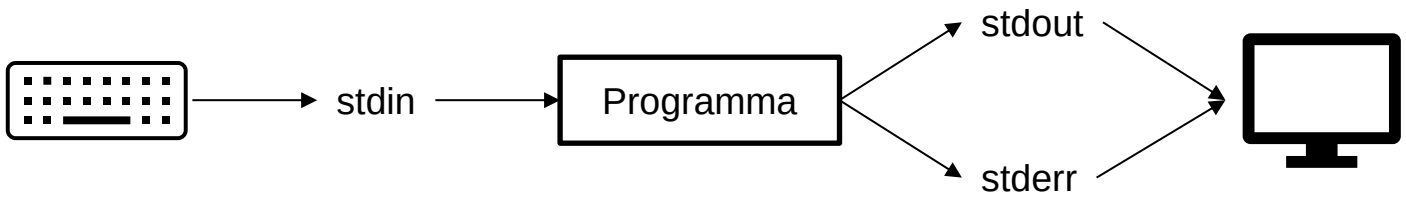
```
corso@corso-os:~$ wc -c combined.txt
54  252  1044       combined.txt
corso@corso-os:~$
```

Altri usi di wc

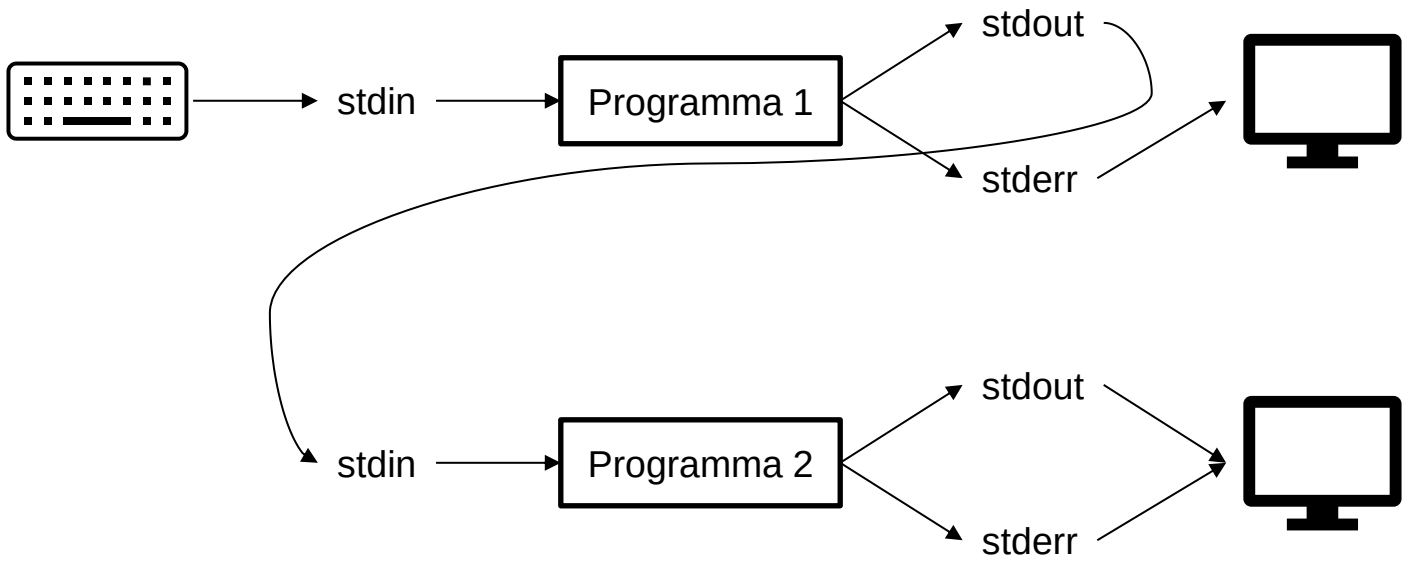
- Supponiamo di voler contare gli elementi (files e directory) contenuti nella nostra home directory, andando successivamente a fare un po' di pulizia. Possiamo utilizzare *wc.*
  - *ls ~ > file_list*
  - *wc –l file_list*
  - *rm file_list*

- Funziona, ma è molto scomodo: ho dovuto eseguire tre comandi!

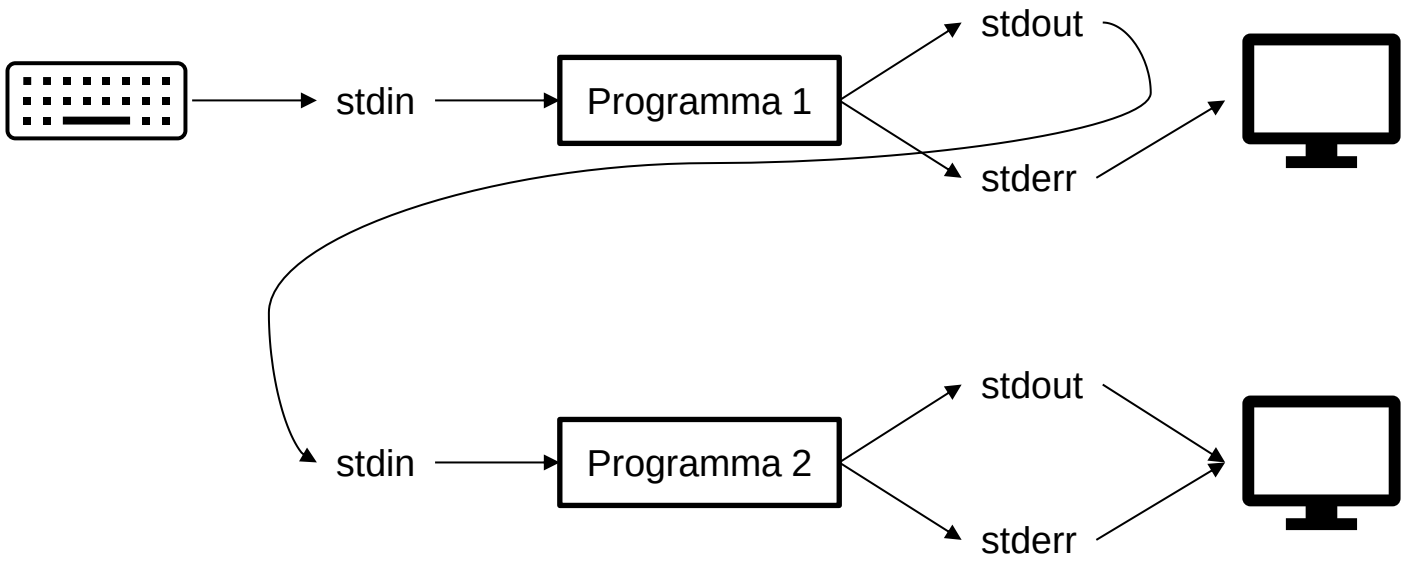
Collegare input e output di comandi



- Abbiamo già visto come redirigere lo standard output su file. Non sarebbe bello poter mandare lo standard output di un primo comando in input ad un secondo comando? Graficamente:

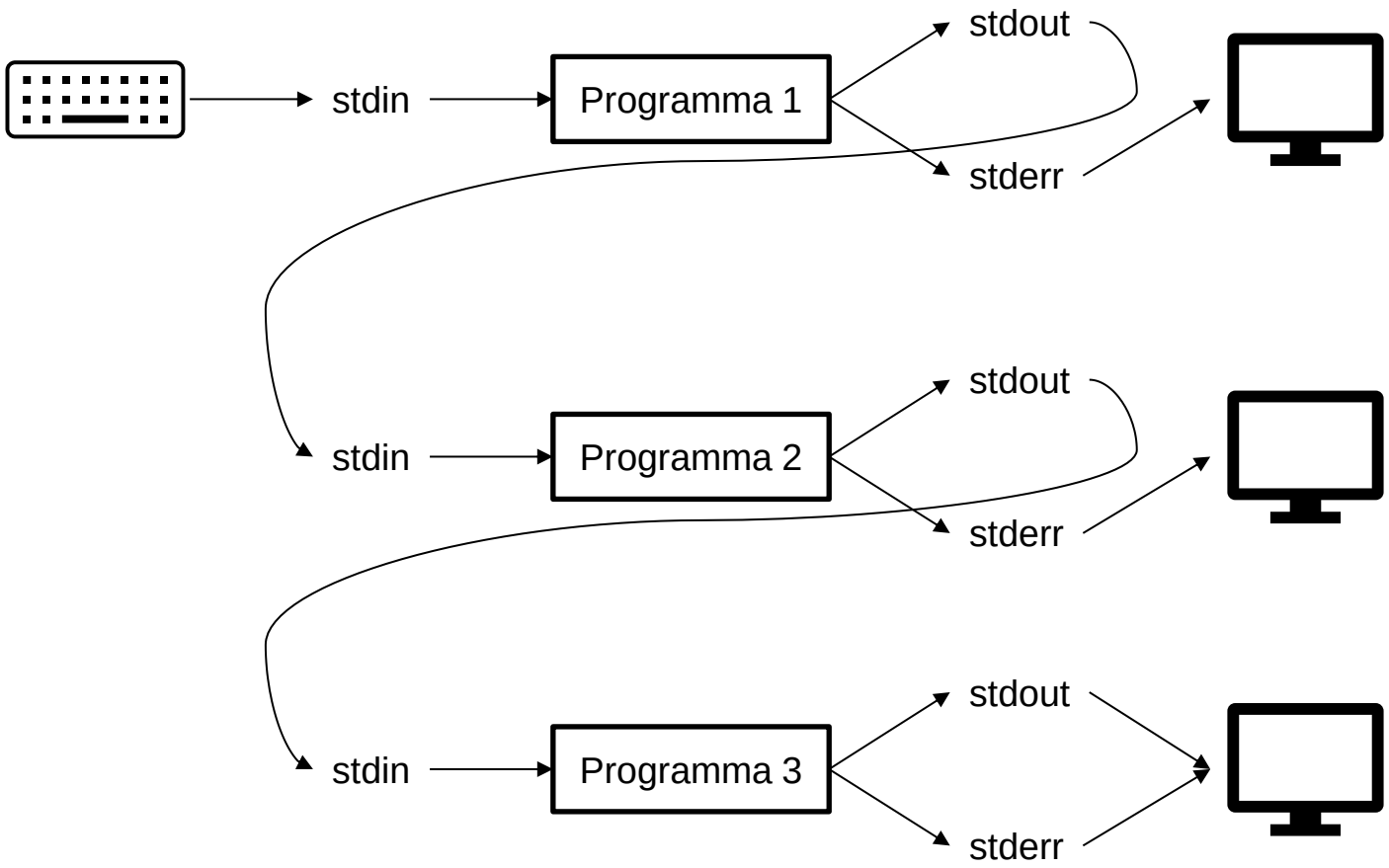


Il meccanismo di piping



- Questo schema è realizzabile tramite un meccanismo di comunicazione chiamato *pipe.*
  - Una *pipe* viene rappresentata dal carattere |
  - Esempio: ls ~ | wc -l
  - Altro esempio: ls /etc | less

Pipe multiple



- Posso replicare questo schema quante volte voglio.
  - Esempio: supponiamo di voler conoscere il numero di righe identiche nel file *combined.txt*

    cat combined.txt | uniq | wc -l

sort, uniq

- Qualcosa non ha funzionato nel comando precedente. Cerchiamo di capire perché

```
corso@corso-os:~$ uniq --help
Usage: uniq [OPTION]... [INPUT [OUTPUT]]
Filter adjacent matching lines from INPUT (or standard input),
writing to OUTPUT (or standard output).
```

- Quindi il problema sta nel fatto che le righe uguali non sono adiacenti, quindi *uniq* non è stato in grado di scovarle. Come possiamo fare per renderle adiacenti?
  - Ci può aiutare il comando *sort*. Proviamo ad eseguire: sort combined.txt | less
- A questo punto, riscriviamo il nostro comando per contare le righe uniche nel file:

```
corso@corso-os:~$ sort combined.txt | uniq | wc -l
3
corso@corso-os:~$
```