

# Sistemi Operativi

## Shell Custom

Gruppo LABSO1-2017\_2018--187803\_187947\_188178

### Caratteristiche e funzionalità avanzate:

#### *Lettura argomenti:*

La shell supporta diversi argomenti, passabili sia in formato esteso che ridotto, in qualsiasi ordine.

#### *History:*

La shell fa uso della libreria **history** per fornire all'utente una cronologia dei comandi digitati, scorrevole attraverso l'uso dei tasti freccia su e giù. Fornisce inoltre la funzionalità di ricerca dei comandi passati attraverso Ctrl-R.

#### *Piping:*

La funzione di piping è implementata ricorsivamente, consentendo l'inoltro dell'output di un comando all'input del successivo, che può a sua volta essere un insieme di pipe.

#### *Autocomplete:*

Il tasto **TAB**, premuto in seguito a dei caratteri, invocherà la funzione di autocomplete, ovvero verrà cercata una corrispondenza tra l'input e la parte iniziale di comandi o file nella cartella corrente, completando la linea di comando con il loro nome completo.

#### *Concatenazione comandi con ; e &&:*

La nostra shell è in grado di distinguere ed eseguire separatamente comandi separati da ; o concatenati con &&.

#### *Esecuzione in background:*

Se una linea di comando viene terminata con &, questa sarà eseguita in un processo figlio completamente indipendente dal padre.

#### *Gestione segnali:*

I segnali **SIGINT** e **SIGALRM** vengono correttamente gestiti. Il primo fa terminare il comando in esecuzione o in alternativa l'intera shell, mentre il secondo viene generato allo scadere di un tetto massimo nel tempo di esecuzione dei processi, causandone la terminazione.

#### *Alias:*

Attraverso il comando **alias** gestito internamente, è possibile definire alias personalizzati sui comandi. La funzionalità è implementata tramite vettori dinamici.

#### *Wildcards \*:*

Se come argomento ad `ls` viene passato un nome con una wildcard “\*” (ad es. \*.txt, a\*, ecc) la shell è in grado di simulare l’espansione della wildcard combinando `ls` ed il comando `grep`.

#### *Variabili:*

Prima dell’esecuzione di un comando, i suoi parametri vengono confrontati con i nomi delle variabili create dall’utente attraverso il comando **var** ed eventualmente sostituite con il corrispettivo valore. In caso una variabile non sia presente nelle definizioni, la ricerca viene estesa alle variabili d’ambiente del terminale da cui è stato lanciato il processo della shell.

#### *Cicli for:*

Se una linea di comando inizia con la parola chiave **for**, il controllo viene passato ad una funzione che ne interpreta variabili e sintassi, fornendone un’implementazione basilare.

#### *Redirect stdout/err/in:*

I canali di output, error ed input possono essere redirezionati su file attraverso i caratteri `>`, `>>`, `&>`, `2>` e `<`;