**Relazione laboratorio Microbash SETI**

**Test semplici**

Per i casi di test più semplici abbiamo scritto uno script test.bash che esegue tutti i comandi contenuti in un file txt su microbash e confronta gli output con quelli ottenuti dagli stessi comandi eseguiti sulla vera bash.

Questo script ha lo scopo di testare in automatico quei comandi che non hanno stampe/gestione degli errori/argomenti personalizzati, come ls, cat, echo, la stampa di variabili d’ambiente o pipe tra questi comandi.

**Variabili d’ambiente non esistenti**

* **Scopo** testare che non venga stampata una variabile d’ambiente non esistente e senza errori, ma solo uno spazio
* **Situazione iniziale** $VariabileCheNonEsiste non esiste come variabile d’ambiente o variabile della sessione del terminale
* **Linea inviata** echo $VariabileCheNonEsiste
* **Risultato** **atteso** spazio vuoto

**Lettura da file senza permessi**

* **Scopo** testare che venga segnalata mancanza di permessi in caso di tentativo di scrittura in file senza permessi di scrittura e che venga segnalata correttamente la terminazione del processo figlio incaricato di svolgere la cat
* **Situazione iniziale** file noPermessi.txt presente in root progetto (senza permessi di lettura)
* **Linea inviata** cat noPermessi.txt
* **Risultato** **atteso** cat: noPermessi.txt: Permission denied Process with PID xxxxx has terminated with exit-status 1

**Scrittura su file senza permessi**

* **Scopo** testare che venga segnalata mancanza di permessi in caso di tentativo di scrittura in file senza permessi di scrittura
* **Situazione iniziale** file noPermessi.txt presente in root progetto (senza permessi di scrittura)
* **Linea inviata** echo ciao >noPermessi.txt
* **Risultato** **atteso** wrong path provided or permission error: Permission denied

**Scrittura su file non esistente**

* **Scopo** verificare che venga creato un file se si vuole scrivere su un file che non esiste
* **Situazione iniziale** file a.txt non esistente
* **Linea inviata** echo ciao >a.txt
* **Risultato** **atteso** nessun errore, creazione del file a.txt contenente la scritta ciao

**CD in pipe**

* **Scopo** verificare che non si possa usare cd in pipe con altri comandi e che venga mostrato l’errore opportuno
* **Situazione iniziale** sottodirectory foo esistente
* **Linea inviata** ls | cd foo
* **Risultato** **atteso** Parsing error: cd command must be the only command of the line

**CD con più di un argomento**

* **Scopo** verificare che cd accetti solo un argomento e che venga mostrato l’errore opportuno
* **Situazione iniziale** sottodirectory foo esistente
* **Linea inviata** cd foo bar
* **Risultato** **atteso** Parsing error: cd command must have only one argument

**CD con redirezione input/output**

* **Scopo** verificare che cd non accetti redirezioni I/O
* **Situazione iniziale** sottodirectory foo esistente
* **Linea inviata** cd foo <bar, cd foo >bar
* **Risultato** **atteso** Parsing error: cd command cannot have I/O redirections

**Spazio tra >/< e nome del file**

* **Scopo** verificare che si possa fare redirezione I/O solo se il nome del file è attaccato ai caratteri >/<
* **Situazione iniziale** //
* **Linea inviata** cat < /proc/cpuinfo | grep processor | wc -l
* **Risultato** **atteso** Parsing error: no path specified for input redirection

**Solo primo comando può avere redirezione input**

* **Scopo** verificare che solo il primo comando supporti la redirezione dell’input
* **Situazione iniziale** //
* **Linea inviata** ls | grep foo <bar | wc -l
* **Risultato** **atteso** Parsing error: only the first command of a line can have input-redirection

**Solo ultimo comando può avere redirezione output**

* **Scopo** verificare che solo l’ultimo comando supporti la redirezione dell’output
* **Situazione iniziale** //
* **Linea inviata** ls -l >bar | grep foo
* **Risultato** **atteso** Parsing error: only the last command of a line can have output-redirection