

Relazione Laboratorio `µbash`

Sara Oliveri (5578081), Fabio Benjamine Roncallo (5601246), Marco Zoratti (5562866).

Tutto il codice è consultabile anche sul Repository GitHub pubblico:

[Uni-Informatica/SecondoAnno/SETI/microbash at master · marchacio/Uni-Informatica \(github.com\)](https://github.com/marchacio/Uni-Informatica)

Testing

Test `parse_cmd`

- Test parsing della variabile d'ambiente "`$PATH`":
 - *Scopo*: controllare che vengano espansi le variabili d'ambiente che iniziano con "`$`"
 - *Situazione iniziale*: `$PATH` esistente
 - *Linea inviata alla microbash*: `echo $PATH`
 - *Risultato*: `/home/marco/.vscode-server/bin/2b35e1e6d88f1ce073683991d1eff5284a32690f/bin/remote-cli:/home/marco/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib`
- Test parsing della variabile inesistente "`$prova`":
 - *Scopo*: controllare che le variabili non inizializzate vengano espansi in una stringa vuota (`""`)
 - *Situazione iniziale*: "`prova`" non è definito
 - *Linea inviata alla microbash*: `echo $prova`
 - *Risultato*: (stringa vuota)
- Test parsing del carattere speciale "`$$`":
 - *Scopo*: controllare che la stringa "`$$`" venga espansa nel pid del processo attuale
 - *Situazione iniziale*: processo attuale: 3733
 - *Linee inviate alla microbash*:
 - `echo $$`
 - `kill $$`
 - *Risultati (rispettivamente)*:
 - 3733
 - Terminated

Test `check_cd`

- Test con altri comandi in coda (stesso risultato con altri comandi in testa):
 - *Scopo*: controllare che il `check_cd` rilevi errori nell'inserimento del comando
 - *Situazione iniziale*: non influente nel test
 - *Linea inviata alla microbash*: `cd bin | ls`
 - *Risultato*: --Errore: Il comando `cd` non può essere utilizzato con altri comandi!
- Test con redirectione I/O:
 - *Scopo*: controllare che il `check_cd` rilevi errori nell'inserimento del comando

- *Situazione iniziale:* “out” e “in” inizializzati, la cartella “bin” esiste;
- *Linee inviate alla microbash:*
 - cd bin >out
 - cd bin <in
- *Risultato:* --Errore: Il comando cd non può redizionare input/output!
- Test con piu parametri:
 - *Scopo:* controllare che il check_cd rilevi errori nell’inserimento del comando
 - *Situazione iniziale:* esistono le cartelle “pippo” e “pluto”;
 - *Linea inviata alla microbash:* cd pippo pluto
 - *Risultato:* --Errore: Il comando cd prende UN SOLO argomento, non 2!

Test change_current_directory

- Test cartella non esistente:
 - *Scopo:* controllare che il cd verso una cartella non esistente non provochi errori
 - *Situazione iniziale:* cartella “pippo” non esistente
 - *Linea inviata alla microbash:* cd pippo
 - *Risultato:* --Errore: la cartella "pippo" non esiste in questa directory
- Test non-cartella:
 - *Scopo:* controllare che il cd verso una non-cartella non provochi errori
 - *Situazione iniziale:* “microbash.c” è un file
 - *Linea inviata alla microbash:* cd microbash.c
 - *Risultato:* --Errore: microbash.c non è una cartella

Test check_redirections

- Test controllo esistenza file di input:
 - *Scopo:* controllare che il file di input esista
 - *Situazione iniziale:* file “prova.txt” non esiste
 - *Linea inviata alla microbash:* cat <prova.txt
 - *Risultato:* --Errore: File prova.txt inesistente!
- Test controllo input:
 - *Scopo:* controllare che il check_redirections rilevi errori nel reindirizzamento dell’input
 - *Situazione iniziale:* non influente nel test
 - *Linea inviata alla microbash:* ls | cat </proc/cpuinfo
 - *Risultato:* --Errore: Solo il primo comando può redizionare l'input!
- Test controllo output:
 - *Scopo:* controllare che il check_redirections rilevi errori nel reindirizzamento dell’output
 - *Situazione iniziale:* non influente nel test
 - *Linea inviata alla microbash:* ls | ls >file.txt | cwd
 - *Risultato:* --Errore: Solo l'ultimo comando può redizionare l'output!

Test execute_line

- Test scrittura in std_out su file inesistente:
 - *Scopo:* controllare che open crei e scriva su un file inesistente
 - *Situazione iniziale:* file “out.txt” inesistente
 - *Linea inviata alla microbash:* echo "ciao" >out.txt
 - *Risultato:* (creazione del file out.txt contenente la stringa “ciao” alla prima riga)

- Test scrittura in std_out su file esistente:
 - *Scopo:* controllare che open apra un file esistente in modalità O_TRUNC
 - *Situazione iniziale:* file "out.txt" esistente, contenente la stringa "ciao"
 - *Linea inviata alla microbash:* echo "seconda prova" >out.txt
 - *Risultato:* (sovrascrittura del file out.txt, contenente "seconda prova")
- Test lettura da std_in da file esistente:
 - *Scopo:* controllare che open apra il file esistente in lettura
 - *Situazione iniziale:* file "out.txt" esistente, contenente la stringa "seconda prova"
 - *Linea inviata alla microbash:* cat <out.txt
 - *Risultato:* seconda prova
- Test lettura da std_in da file inesistente non effettuato in quanto è un errore rilevato da [check_redirections](#)
- Test reindirizzamento I/O tramite pipe:
 - *Scopo:* controllare che pipe prenda l'output del primo comando e lo mandi al secondo come input.
 - *Situazione iniziale:* file nella cartella attuale: microbash.o, microbash.c, microbash, makefile, Relazione.pdf
 - *Linea inviata alla microbash:* ls | grep micro
 - *Risultato:* microbash \n microbash.c \n microbash.o

Test run_child

- Test esecuzione comandi:
 - *Scopo:* controllare che execvp esegua correttamente i comandi della linea e gestisca correttamente gli errori possibili.
 - *Situazione iniziale:* comando "cwd" non installato, file ./prova non esistente
 - *Linee inviate alla microbash:*
 - cwd
 - ./prova
 - *Risultati (rispettivamente):*
 - Errore: file/comando non esistente e/o permessi mancanti
-----Exit status: 1-----
 - Errore: il comando/file non esiste
-----Exit status: 1-----
- Test esecuzione comandi con parametri
 - *Scopo:* controllare che l'esecuzione del comando rispetti gli argomenti passati
 - *Situazione iniziale:* Cartella contenente microbash.pdf, microbash.o, microbash, Relazione.docx, microbash.c, Makefile.
 - *Linea inviata alla microbash:* ls -l -h -S
 - *Risultato:* total 1.5M
 - rwxrwxrwx 1 marco marco 1.4M Nov 8 12:30 microbash.pdf
 - rwxrwxrwx 1 marco marco 67K Nov 10 06:46 microbash.o
 - rwxrwxrwx 1 marco marco 57K Nov 10 06:46 microbash
 - rwxrwxrwx 1 marco marco 19K Nov 9 11:17 'Relazione.docx'
 - rwxrwxrwx 1 marco marco 17K Nov 10 06:52 microbash.c
 - rwxrwxrwx 1 marco marco 320 Nov 9 19:41 Makefile

Test wait_for_children

- Test attesa termine esecuzione:
 - *Scopo*: controllare che la µbash attenda la fine dell'esecuzione di un comando
 - *Situazione iniziale*: non influente nel test
 - *Linea inviata alla microbash*: sleep 10
 - *Risultato*: (microbash ha atteso la fine dell'esecuzione del comando correttamente)
- Test uccisione di processo da segnale:
 - *Scopo*: controllare che, in caso di interruzione da segnale, il processo termini e venga mostrato l'errore
 - *Situazione iniziale*: non influente nel test
 - *Linea inviata alla microbash*: sleep 1000
 - *Operazioni extra*: apertura del programma "htop" in un secondo terminale per uccidere il processo "sleep"
 - *Risultato*: --Errore: Processo ucciso da segnale (esecuzione del comando sleep terminata)
- Test EXITSTATUS:
 - *Scopo*: controllare che la µbash stampi un messaggio di errore contenente l'EXITSTATUS quando l'esecuzione di un programma fallisce
 - *Situazione iniziale*: comando "cwd" non installato e pagina man WIFSIGNALED non esistente
 - *Linee inviate alla microbash*:
 - cwd
 - man WIFSIGNALED
 - *Risultati (rispettivamente)*:
 - --Errore: file/comando non esistente e/o permessi mancanti
-----Exit status: 1-----
 - No manual entry for WIFSIGNALED
-----Exit status: 16-----

Ambiente di sviluppo, compilazione ed esecuzione

Tutto il codice è stato sviluppato su WSL2 (Ubuntu 22.04) con l'ausilio di Visual Studio Code e compilato con il comando *make all* dalla cartella principale del progetto.

Versione compilatore CC: 11.4.0.

Archivio Tar Gz creato tramite il comando: *tar cvJf microbash-students.tar.xz microbash-students*