





# REPORT TEST MINISHELL

## PANORAMICA GENERALE

Questo documento riporta i risultati dei test eseguiti sulla minishell, suddivisi per categorie funzionali. Ogni sezione include i test eseguiti, i risultati ottenuti e le note implementative.

### 1. ESECUZIONE COMANDI BASE















Test Completati con Successo:

Comando	Risultato	Note
foo	 → foo command not found	Gestione corretta comando inesistente
/bin/ls		Esecuzione path assoluto
/bin/ls -laF		Supporto argomenti multipli
/bin/ls -l -a -F		Argomenti separati






Status:  COMPLETATA CON SUCCESSO

### 2. BUILT-IN COMMANDS






#### 2.1 Exit Command

Test	Comando	Risultato Atteso	Risultato Ottenuto	Status
Exit semplice	exit	Chiusura pulita (0)		
Exit con codice	exit 42	Chiusura (status 42)		
Exit argomento non numerico	exit hello	Errore + status 255		
Exit troppi argomenti	exit 1 2	Errore, shell continua		
Exit numero negativo	exit -1	Chiusura (status 255)		
Exit overflow	exit 999999...	Errore + status 255		
Exit modulo 256	exit 258	Chiusura (status 2)		

#### 2.2 Echo Command

Test	Comando	Risultato	Status
Echo con virgolette	echo "It works"	It works	
Echo senza virgolette	echo it works	it works	
Echo con flag -n	echo -n ciao	ciao (senza newline)	
Echo -n multipli	echo -n -n ciao	ciao (senza newline)	
Echo -n con altri flag	echo -n -e ciao	-e ciao (senza newline)	

#### 2.3 Change Directory (cd)

Test	Comando	Risultato	Status
CD path assoluto	cd /tmp	Cambio directory corretto	
CD path relativo	cd src	Cambio directory corretto	
CD senza argomenti	cd	Ritorno a HOME	
CD con dash	cd -	OLDPWD not set	
CD con tilde	cd ~/Desktop	No such file or directory	

#### Problemi Identificati:

- Gestione di OLDPWD non implementata
- Espansione della tilde ~ non supportata

### 3. GESTIONE DELL'AMBIENTE

#### 3.1 Variabili d'Ambiente

Test	Comando	Risultato	Status
Visualizza env	env	Lista completa variabili	✓
Export variabile	export FOO=bar	Variabile aggiunta	✓
Echo variabile	echo \$FOO	bar	✓
Env esterno	/usr/bin/env	Variabili non aggiornate	✗
Unset variabile	unset FOO	Variabile rimossa	✓

#### PROBLEMA CRITICO:

Le variabili esportate non vengono passate correttamente ai processi figli tramite envp in execve()

#### 3.2 Gestione PATH

Test	Comando	Risultato	Status
Export PATH custom	export PATH="/bin:/usr/bin" + ls	Funziona	✓
Unset PATH	unset PATH + ls	Ancora funziona (err.)	✗
Path assoluto	/bin/ls	Funziona	✓
Comando in /usr/bin	/usr/bin/wc Makefile	Funziona	✓

#### Problema:

unset PATH non impedisce la ricerca nei path di default

### 4. PARSING E INPUT

#### 4.1 Gestione Spazi e Input

Test	Comando	Risultato	Status
Input vuoto	[Enter]	Nessun output, nuovo prompt	✓
Spazi multipli	echo    ciao    mondo	ciao mondo	✓
Virgolette doppie	echo "ciao    mondo"	Preserva spazi	✓
Virgolette singole	echo 'ciao mondo'	ciao mondo	✓

#### 4.2 Espansione Variabili

Test	Comando	Risultato	Status
Variabile in doppie virgolette	echo "\$VAR"	Espansione corretta	✓
Variabile inesistente	echo \$NONEXISTENT	Output vuoto	✓
Variabile in singole virgolette	echo '\$NONEXISTENT'	Letterale	✓
Escape sequences	echo "ciao\nmondo"	Nessuna interpretazione	✓

## 5. REDIREZIONE E PIPE

### 5.1 Redirezione Output

Test	Comando	Risultato	Status
Redirezione semplice	echo hello > test.txt	File creato	✓
Append	echo world >> test.txt	Aggiunto al file	✓

### 5.2 Pipeline

Test	Comando	Risultato	Status
Pipe semplice	ls   cat	Output corretto	✓
Pipe con filtro	cat Makefile   grep minishell	Filtro funziona	✓
Pipe con conteggio	ls -l   wc -l	Conta righe	✓
Pipe con char count	echo hello   wc -c	6 (con newline)	✓

### 5.3 Redirezione Input (Non Implementata)

Test	Comando	Risultato	Status
Input redirect	cat < test.txt	Parsing errato	✗

Nota: La redirezione input e heredoc non sono ancora implementate.

## 6. GESTIONE ERRORI E MEMORY MANAGEMENT

### 6.1 Memory Leaks (Valgrind)

- ✓ ECCELLENTE GESTIONE MEMORIA:
- Leak summary sempre: definitely lost: 0 bytes in 0 blocks
  - Solo leak "suppressed" da Valgrind (normali)
  - Nessun errore di memoria rilevato

### 6.2 Gestione Errori

Scenario	Comportamento	Status
Comando inesistente	Messaggio d'errore appropriato	✓
Argomenti invalidi per exit	Messaggio + exit code corretto	✓
Directory inesistente (cd)	Messaggio d'errore	✓
File inesistente	Gestito dal comando esterno	✓

## RIASSUNTO FUNZIONALITA'

### ✓ IMPLEMENTATE E FUNZIONANTI:

- Esecuzione comandi con path assoluto
- Built-in: echo, exit, cd (base), env, export, unset
- Gestione variabili d'ambiente (interno)
- Parsing avanzato (virgolette, spazi, espansione variabili)
- Pipeline semplici e multiple
- Redirezione output (>, >>)
- Gestione memoria eccellente

### ✗ DA IMPLEMENTARE/CORREGGERE:

- CRITICO: Passaggio variabili a processi figli (envp in execve)
- IMPORTANTE: Gestione OLDPWD per cd -
- IMPORTANTE: Espansione tilde ~ per cd
- IMPORTANTE: Rispetto di unset PATH
- FEATURE: Redirezione input (<)
- FEATURE: Heredoc (<<)

### ■ COMPORTAMENTI PARTICOLARI (CONFORMI):

- echo -e non interpreta escape (scelta implementativa valida)
- echo senza flag -n è il comportamento standard
- Variabili inesistenti espandono a stringa vuota

=====

DATA REPORT: 8 Agosto 2025

TOOL UTILIZZATI: Valgrind, test manuali

AMBIENTE: Ubuntu Linux