

# Contents

<b>1</b>	<b>Guide d'Utilisation et Exemples Minishell</b>	<b>2</b>
1.1	Table des Matières . . . . .	2
1.2	Installation et Compilation . . . . .	2
1.2.1	Prérequis . . . . .	2
1.2.2	Compilation . . . . .	2
1.2.3	Vérification de l'Installation . . . . .	3
1.3	Commandes de Base . . . . .	3
1.3.1	Exécution de Programmes . . . . .	3
1.3.2	Navigation . . . . .	3
1.4	Redirections . . . . .	4
1.4.1	Redirection de Sortie . . . . .	4
1.4.2	Redirection d'Entrée . . . . .	4
1.4.3	Here Document (Heredoc) . . . . .	5
1.5	Pipes et Pipelines . . . . .	6
1.5.1	Pipes Simples . . . . .	6
1.5.2	Pipes Multiples . . . . .	6
1.5.3	Combinaison Pipes et Redirections . . . . .	6
1.6	Opérateurs Logiques . . . . .	7
1.6.1	Opérateur AND (&&) . . . . .	7
1.6.2	Opérateur OR (  ) . . . . .	7
1.6.3	Combinaison d'Opérateurs . . . . .	7
1.7	Variables d'Environnement . . . . .	8
1.7.1	Affichage des Variables . . . . .	8
1.7.2	Définition de Variables . . . . .	8
1.7.3	Modification de Variables Système . . . . .	8
1.7.4	Suppression de Variables . . . . .	9
1.7.5	Variables Spéciales . . . . .	9
1.8	Commandes Intégrées . . . . .	9
1.8.1	echo . . . . .	9
1.8.2	pwd . . . . .	10
1.8.3	cd . . . . .	10
1.8.4	env . . . . .	10
1.8.5	export . . . . .	11
1.8.6	unset . . . . .	11
1.8.7	exit . . . . .	12
1.9	Wildcards et Expansion . . . . .	12
1.9.1	Expansion d'Étoile (*) . . . . .	12
1.9.2	Expansion de Point d'Interrogation (?) . . . . .	12
1.9.3	Expansion de Tilde (~) . . . . .	13
1.9.4	Combinaisons d'Expansions . . . . .	13
1.10	Exemples Avancés . . . . .	13
1.10.1	Exemple 1 : Script de Sauvegarde . . . . .	13
1.10.2	Exemple 2 : Analyse de Logs . . . . .	13
1.10.3	Exemple 3 : Développement C . . . . .	14
1.10.4	Exemple 4 : Traitement de Fichiers . . . . .	14
1.10.5	Exemple 5 : Monitoring Système . . . . .	15
1.11	Troubleshooting . . . . .	15
1.11.1	Problèmes Courants . . . . .	15

1.11.2	Debugging . . . . .	16
1.11.3	Optimisation des Performances . . . . .	17
1.12	Conclusion . . . . .	17
1.12.1	Fonctionnalités Clés . . . . .	17
1.12.2	Ressources Additionnelles . . . . .	17

# 1 Guide d'Utilisation et Exemples Minishell

## 1.1 Table des Matières

1. [Installation et Compilation](#)
  2. [Commandes de Base](#)
  3. [Redirections](#)
  4. [Pipes et Pipelines](#)
  5. [Opérateurs Logiques](#)
  6. [Variables d'Environnement](#)
  7. [Commandes Intégrées](#)
  8. [Wildcards et Expansion](#)
  9. [Exemples Avancés](#)
  10. [Troubleshooting](#)
- 

## 1.2 Installation et Compilation

### 1.2.1 Prérequis

```
# Ubuntu/Debian
sudo apt-get update
sudo apt-get install build-essential libreadline-dev

# macOS avec Homebrew
brew install readline
export CPPFLAGS="-I/usr/local/opt/readline/include"
export LDFLAGS="-L/usr/local/opt/readline/lib"

# Arch Linux
sudo pacman -S readline
```

### 1.2.2 Compilation

```
# Clone du projet
git clone [url-du-projet]
cd minishell

# Compilation standard
make

# Compilation avec debug
make clean
```

```
# Décommenter les lignes ASAN dans le Makefile
make
```

```
# Nettoyage
make clean      # Supprime objets
make fclean     # Supprime tout
make re         # Recompilation complète
```

### 1.2.3 Vérification de l'Installation

```
# Test de base
./minishell
minishell> echo "Hello World"
Hello World
minishell> exit

# Test avec Valgrind
valgrind --leak-check=full --suppressions=readline.supp ./minishell
```

---

## 1.3 Commandes de Base

### 1.3.1 Exécution de Programmes

```
# Commandes système standard
minishell> ls
minishell> ls -la
minishell> cat /etc/passwd
minishell> grep root /etc/passwd

# Chemins absolus
minishell> /bin/ls
minishell> /usr/bin/whoami

# Chemins relatifs
minishell> ./mon_script.sh
minishell> ../autre_repertoire/programme
```

### 1.3.2 Navigation

```
# Affichage répertoire courant
minishell> pwd
/home/user/project

# Changement de répertoire
minishell> cd /tmp
minishell> pwd
/tmp
```

```
# Retour au répertoire précédent
```

```
minishell> cd -  
/home/user/project
```

```
# Répertoire home
```

```
minishell> cd  
minishell> pwd  
/home/user
```

```
# Répertoire parent
```

```
minishell> cd ..  
minishell> pwd  
/home
```

---

## 1.4 Redirections

### 1.4.1 Redirection de Sortie

```
# Redirection simple (écrase le fichier)
```

```
minishell> echo "Hello" > output.txt  
minishell> cat output.txt  
Hello
```

```
# Redirection en mode append (ajoute au fichier)
```

```
minishell> echo "World" >> output.txt  
minishell> cat output.txt  
Hello  
World
```

```
# Redirection avec commande complexe
```

```
minishell> ls -la > liste_fichiers.txt  
minishell> wc -l liste_fichiers.txt  
15 liste_fichiers.txt
```

### 1.4.2 Redirection d'Entrée

```
# Créer un fichier de test
```

```
minishell> echo -e "ligne1\nligne2\nligne3" > input.txt
```

```
# Redirection d'entrée
```

```
minishell> cat < input.txt  
ligne1  
ligne2  
ligne3
```

```
# Avec processing
```

```
minishell> sort < input.txt
```

```
ligne1
ligne2
ligne3

# Redirection combinée
minishell> sort < input.txt > sorted.txt
minishell> cat sorted.txt
ligne1
ligne2
ligne3
```

### 1.4.3 Here Document (Heredoc)

```
# Heredoc simple
minishell> cat << EOF
> Première ligne
> Deuxième ligne
> Troisième ligne
> EOF
Première ligne
Deuxième ligne
Troisième ligne

# Heredoc avec redirection
minishell> cat << END > multi_lines.txt
> Line 1
> Line 2
> Line 3
> END
minishell> cat multi_lines.txt
Line 1
Line 2
Line 3

# Heredoc avec variables
minishell> export NAME="John"
minishell> cat << EOF
> Hello $NAME
> Your home is $HOME
> EOF
Hello John
Your home is /home/user
```

---

## 1.5 Pipes et Pipelines

### 1.5.1 Pipes Simples

```
# Pipe basique
minishell> ls | wc -l
25

# Filtrage avec grep
minishell> ls -la | grep ".txt"
-rw-r--r-- 1 user user  42 Jan  1 12:00 file.txt
-rw-r--r-- 1 user user 156 Jan  1 12:01 notes.txt

# Tri de résultats
minishell> ps aux | sort
# Liste des processus triée
```

### 1.5.2 Pipes Multiples

```
# Triple pipe
minishell> ls -la | grep ".c" | wc -l
12

# Pipeline complexe
minishell> cat /etc/passwd | cut -d: -f1 | sort | uniq
# Liste des utilisateurs triée et unique

# Avec head et tail
minishell> ps aux | head -20 | tail -10
# Lignes 11 à 20 de ps aux

# Pipeline avec redirection
minishell> ls -la | grep ".txt" | sort > fichiers_txt_tries.txt
```

### 1.5.3 Combinaison Pipes et Redirections

```
# Input et output avec pipes
minishell> cat < input.txt | sort | uniq > output_unique.txt

# Pipeline avec heredoc
minishell> cat << EOF | grep "important" | wc -l
> ligne normale
> ligne importante
> autre ligne importante
> ligne finale
> EOF
2
```

## 1.6 Opérateurs Logiques

### 1.6.1 Opérateur AND (&&)

```
# Exécution conditionnelle (succès)
minishell> echo "test" && echo "success"
test
success

# Exécution conditionnelle (échec)
minishell> false && echo "not executed"
# "not executed" n'est pas affiché

# Chaîne de commandes
minishell> mkdir test_dir && cd test_dir && pwd
/current/path/test_dir

# Avec redirections
minishell> echo "data" > file.txt && cat file.txt
data
```

### 1.6.2 Opérateur OR (||)

```
# Alternative en cas d'échec
minishell> false || echo "fallback executed"
fallback executed

# Avec commande qui réussit
minishell> true || echo "not executed"
# "not executed" n'est pas affiché

# Création de répertoire avec fallback
minishell> mkdir existing_dir || echo "Directory already exists"
mkdir: existing_dir: File exists
Directory already exists
```

### 1.6.3 Combinaison d'Opérateurs

```
# AND puis OR
minishell> false && echo "not shown" || echo "fallback"
fallback

# OR puis AND
minishell> true || echo "not shown" && echo "always shown"
always shown

# Groupement avec parenthèses
minishell> (echo "first" && echo "second") || echo "fallback"
first
second
```

```
# Commandes complexes
minishell> ls file.txt && cat file.txt || echo "File not found"
```

---

## 1.7 Variables d'Environnement

### 1.7.1 Affichage des Variables

```
# Toutes les variables
minishell> env
PATH=/usr/bin:/bin:/usr/sbin:/sbin
HOME=/home/user
USER=user
SHELL=/bin/bash
...

# Variable spécifique
minishell> echo $HOME
/home/user

# Plusieurs variables
minishell> echo "User: $USER, Home: $HOME"
User: user, Home: /home/user
```

### 1.7.2 Définition de Variables

```
# Export d'une variable
minishell> export MY_VAR="Hello World"
minishell> echo $MY_VAR
Hello World

# Variable temporaire
minishell> export TEMP_VAR="temporary"
minishell> env | grep TEMP_VAR
TEMP_VAR=temporary

# Variable avec espaces
minishell> export MESSAGE="Hello World with spaces"
minishell> echo "$MESSAGE"
Hello World with spaces
```

### 1.7.3 Modification de Variables Système

```
# Modification du PATH
minishell> export PATH="$PATH:/nouveau/chemin"
minishell> echo $PATH
/usr/bin:/bin:/usr/sbin:/sbin:/nouveau/chemin
```



```
# Changement du PS1 (si supporté)
minishell> export PS1="custom> "

# Variables pour compilation
minishell> export CFLAGS="-Wall -Wextra -Werror"
minishell> export CC="gcc"
```

#### 1.7.4 Suppression de Variables

```
# Définir une variable
minishell> export TEST_VAR="to be deleted"
minishell> echo $TEST_VAR
to be deleted

# Supprimer la variable
minishell> unset TEST_VAR
minishell> echo $TEST_VAR

# Aucune sortie

# Vérification
minishell> env | grep TEST_VAR

# Aucun résultat
```

#### 1.7.5 Variables Spéciales

```
# Code de retour de la dernière commande
minishell> true
minishell> echo $?
0

minishell> false
minishell> echo $?
1

# Avec commandes complexes
minishell> ls nonexistent_file
ls: nonexistent_file: No such file or directory
minishell> echo $?
2
```

---

### 1.8 Commandes Intégrées

#### 1.8.1 echo

```
# Echo simple
minishell> echo Hello World
Hello World
```

```
# Echo avec option -n (pas de newline)
minishell> echo -n "No newline"
No newline%

# Avec variables
minishell> export NAME="Alice"
minishell> echo "Hello $NAME"
Hello Alice

# Caractères spéciaux
minishell> echo "Quotes: 'single' and \"double\""
Quotes: 'single' and "double"
```

### 1.8.2 pwd

```
# Répertoire courant
minishell> pwd
/home/user/Documents/minishell

# Après changement de répertoire
minishell> cd /tmp
minishell> pwd
/tmp
```

### 1.8.3 cd

```
# Changements de répertoire
minishell> cd /home
minishell> pwd
/home

# Répertoire home
minishell> cd
minishell> pwd
/home/user

# Répertoire précédent
minishell> cd /tmp
minishell> cd -
/home/user

# Répertoire inexistant
minishell> cd /nonexistent
cd: /nonexistent: No such file or directory
```

### 1.8.4 env

```

# Affichage complet
minishell> env
PATH=/usr/bin:/bin
HOME=/home/user
USER=user
...

# Filtrage avec grep
minishell> env | grep PATH
PATH=/usr/bin:/bin:/usr/sbin:/sbin

# Tri
minishell> env | sort

```

### 1.8.5 export

```

# Export simple
minishell> export VAR="value"

# Export multiple
minishell> export VAR1="value1" VAR2="value2"

# Export avec affichage
minishell> export
declare -x HOME="/home/user"
declare -x PATH="/usr/bin:/bin"
declare -x VAR="value"
...

# Export invalide
minishell> export 123VAR="invalid"
export: `123VAR': not a valid identifier

```

### 1.8.6 unset

```

# Supprimer une variable
minishell> export TEST="value"
minishell> echo $TEST
value
minishell> unset TEST
minishell> echo $TEST
# Aucune sortie

# Supprimer plusieurs variables
minishell> export A="1" B="2" C="3"
minishell> unset A B C
minishell> echo $A $B $C
# Aucune sortie

```

### 1.8.7 exit

```
# Sortie normale
minishell> exit
# Shell se ferme avec code 0

# Sortie avec code
minishell> exit 42
# Shell se ferme avec code 42

# Sortie avec argument invalide
minishell> exit invalid
exit: invalid: numeric argument required
# Shell se ferme avec code 2
```

---

## 1.9 Wildcards et Expansion

### 1.9.1 Expansion d'Étoile (\*)

```
# Tous les fichiers
minishell> ls *
# Liste tous les fichiers du répertoire courant

# Fichiers avec extension
minishell> ls *.txt
file1.txt  file2.txt  notes.txt

# Avec sous-répertoires
minishell> ls */*.c
src/main.c  src/utils.c  tests/test.c

# Début de nom
minishell> ls test*
test.txt  test_file.dat  test_script.sh
```

### 1.9.2 Expansion de Point d'Interrogation (?)

```
# Un caractère
minishell> ls file?.txt
file1.txt  file2.txt  fileA.txt

# Plusieurs caractères
minishell> ls test_???.dat
test_001.dat  test_abc.dat  test_xyz.dat

# Combinaison avec étoile
minishell> ls file?.*
file1.txt  file2.dat  fileA.log
```

### 1.9.3 Expansion de Tilde (~)

```
# Répertoire home
minishell> echo ~
/home/user

# Avec chemin
minishell> ls ~/Documents
# Liste le contenu de /home/user/Documents

# Dans commandes
minishell> cd ~/Desktop
minishell> pwd
/home/user/Desktop
```

### 1.9.4 Combinaisons d'Expansions

```
# Tilde et wildcards
minishell> ls ~/*.txt
/home/user/file1.txt  /home/user/notes.txt

# Variables et wildcards
minishell> export DIR="/home/user/Documents"
minishell> ls $DIR/*.pdf
/home/user/Documents/doc1.pdf  /home/user/Documents/manual.pdf

# Wildcards dans pipes
minishell> cat *.txt | grep "pattern"
# Recherche dans tous les fichiers .txt
```

---

## 1.10 Exemples Avancés

### 1.10.1 Exemple 1 : Script de Sauvegarde

```
# Création d'un répertoire de sauvegarde
minishell> mkdir -p ~/backup/$(date +%Y%m%d) && echo "Backup directory created"

# Copie des fichiers importants
minishell> cp *.txt ~/backup/$(date +%Y%m%d)/ && echo "Files backed up"

# Vérification
minishell> ls ~/backup/$(date +%Y%m%d)/ | wc -l && echo "files backed up"
```

### 1.10.2 Exemple 2 : Analyse de Logs

```
# Création d'un fichier de log simulé
minishell> cat << EOF > access.log
> 192.168.1.1 - GET /index.html 200
```

```

> 192.168.1.2 - GET /about.html 404
> 192.168.1.1 - POST /login 200
> 192.168.1.3 - GET /index.html 200
> 192.168.1.2 - GET /contact.html 500
> EOF

# Analyse des codes d'erreur
minishell> grep "404\|500" access.log | wc -l
2

# IPs uniques
minishell> cut -d' ' -f1 access.log | sort | uniq
192.168.1.1
192.168.1.2
192.168.1.3

# Requêtes par IP
minishell> cut -d' ' -f1 access.log | sort | uniq -c | sort -nr
      2 192.168.1.1
      2 192.168.1.2
      1 192.168.1.3

```

### 1.10.3 Exemple 3 : Développement C

```

# Configuration de l'environnement
minishell> export CC="gcc"
minishell> export CFLAGS="-Wall -Wextra -Werror -std=c99"

# Compilation
minishell> $CC $CFLAGS -o program *.c && echo "Compilation successful"

# Test
minishell> ./program && echo "Program executed successfully" || echo "Program failed"

# Nettoyage
minishell> rm -f *.o program && echo "Cleaned up"

```

### 1.10.4 Exemple 4 : Traitement de Fichiers

```

# Créer des fichiers de test
minishell> for i in 1 2 3; do echo "Content $i" > file$i.txt; done

# Concaténation avec numérotation
minishell> cat file*.txt | nl > combined.txt

# Statistiques
minishell> wc -l file*.txt
      1 file1.txt
      1 file2.txt

```

```
1 file3.txt
3 total
```

*# Recherche dans tous les fichiers*

```
minishell> grep -n "Content" file*.txt
file1.txt:1:Content 1
file2.txt:1:Content 2
file3.txt:1:Content 3
```

### 1.10.5 Exemple 5 : Monitoring Système

*# Processus utilisant le plus de CPU*

```
minishell> ps aux | sort -k3 -nr | head -5
```

*# Espace disque*

```
minishell> df -h | grep -v "tmpfs" | sort -k5 -nr
```

*# Mémoire disponible*

```
minishell> free -h | grep "Mem:" | awk '{print "Used: " $3 " , Free: " $7}'
```

*# Services actifs (si disponible)*

```
minishell> ps aux | grep -v grep | grep "daemon\|service" | wc -l
```

---

## 1.11 Troubleshooting

### 1.11.1 Problèmes Courants

*# Erreur*

```
minishell> mycommand
minishell: mycommand: command not found
```

*# Solutions*

*# Vérifier le PATH*

```
minishell> echo $PATH
/usr/bin:/bin:/usr/sbin:/sbin
```

*# Ajouter au PATH*

```
minishell> export PATH="$PATH:/chemin/vers/commande"
```

*# Utiliser chemin absolu*

```
minishell> /chemin/complet/vers/mycommand
```

#### 1.11.1.1 1. Commande Introuvable

*# Erreur*

```
minishell> ./script.sh
minishell: ./script.sh: Permission denied
```

```
# Solution
minishell> chmod +x script.sh
minishell> ./script.sh
```

#### 1.11.1.2 2. Permission Refusée

```
# Erreur
minishell> cat nonexistent.txt
cat: nonexistent.txt: No such file or directory
```

```
# Vérification
minishell> ls -la *.txt
minishell> find . -name "*.txt"
```

#### 1.11.1.3 3. Fichier Introuvable

```
# Erreur avec répertoire
minishell> echo "test" > /nonexistent/file.txt
bash: /nonexistent/file.txt: No such file or directory

# Créer le répertoire d'abord
minishell> mkdir -p /path/to/directory && echo "test" > /path/to/directory/file.txt
```

#### 1.11.1.4 4. Problèmes de Redirections

### 1.11.2 Debugging

```
# Voir les commandes exécutées
minishell> set -x # Si supporté
minishell> echo "test"
+ echo test
test

# Variables d'environnement
minishell> env | grep VAR
minishell> echo $VAR
```

#### 1.11.2.1 Trace d'Exécution

```
# Parenthèses non fermées
minishell> (echo "test"
# Erreur de syntaxe

# Quotes non fermées
minishell> echo "test
# Erreur de syntaxe
```



```
# Pipes mal formés
minishell> ls |
# Erreur de syntaxe
```

### 1.11.2.2 Vérification de Syntaxe

### 1.11.3 Optimisation des Performances

```
# Éviter les pipes inutiles
# Au lieu de:
minishell> cat file.txt | grep pattern

# Utiliser:
minishell> grep pattern file.txt

# Limiter la sortie
minishell> find / -name "*.txt" 2>/dev/null | head -10

# Utiliser des wildcards efficaces
minishell> ls specific_*.txt # Plus efficace que ls * | grep specific_
```

---

## 1.12 Conclusion

Ce guide couvre l'utilisation complète du Minishell, des commandes de base aux cas d'usage avancés. Le shell supporte la plupart des fonctionnalités standard d'un shell Unix, permettant une utilisation productive pour le développement et l'administration système.

### 1.12.1 Fonctionnalités Clés

**Supportées :** - Commandes externes et builtins - Pipes et redirections - Variables d'environnement - Opérateurs logiques - Wildcards et expansion - Historique avec readline - Gestion des signaux

**Non Supportées :** - Job control (bg, fg, jobs) - Alias - Fonctions shell - Boucles (for, while) - Conditions (if, case) - Substitution de commandes

### 1.12.2 Ressources Additionnelles

- **Manual Pages :** `man bash` pour référence complète
- **Standards POSIX :** Spécifications shell POSIX
- **Documentation Readline :** Pour fonctionnalités d'édition
- **Tests :** Utilisez les scripts de test fournis

Ce minishell constitue une excellente base pour comprendre le fonctionnement interne des shells Unix et peut être étendu selon les besoins spécifiques.