

## MyShell

Commands supported <command>	Options supported <option>
echo	<ul style="list-style-type: none"><li>• &lt;any input&gt;</li><li>• -n &lt;any input&gt;</li><li>• - -help</li></ul>
pwd	<ul style="list-style-type: none"><li>• -L</li><li>• -P</li></ul>
cd	<ul style="list-style-type: none"><li>• &lt;directory name&gt;</li><li>• ~</li><li>• &lt;path of directory&gt;</li></ul>
ls	<ul style="list-style-type: none"><li>• -l</li><li>• -r</li></ul>
mkdir	<ul style="list-style-type: none"><li>• &lt;directory name&gt;</li><li>• -v &lt;directory name&gt;</li><li>• -p &lt;directory name&gt;</li></ul>
cat	<ul style="list-style-type: none"><li>• -n &lt;filename&gt;</li><li>• -b &lt;filename&gt;</li></ul>
rm	<ul style="list-style-type: none"><li>• &lt;filename&gt;</li><li>• -v &lt;filename&gt;</li><li>• -r &lt;filename / directory name&gt;</li></ul>
date	<ul style="list-style-type: none"><li>• -u</li><li>• -R</li></ul>
touch	touch <filename>

- Commands executed as <command> <option>

Errors handled:

1. Errors common to all commands
  - Checked if command is valid or not
  - Checked if fork failed or not
  - Checked if pthread\_create, pthread\_join failed or not
  - Checked if flag is valid or not (in all except cd)
2. Errors handled specific to commands

Command	Error
cd	<ul style="list-style-type: none"><li>• Checked if directory exists or not.</li><li>• Checked if permitted to change to that directory or not(eg: cd /root).</li></ul>
mkdir	<ul style="list-style-type: none"><li>• Checked if operands are missing or not</li><li>• Checked if permitted to make directory. (eg: mkdir /root/dir1)</li></ul>
cat	<ul style="list-style-type: none"><li>• Checked if operands are missing or not.</li><li>• Checked if the operand exists, and if it exists is it a file or not.</li><li>• Checked if user has permission to open file or not. (eg: cat /etc/shadow)</li></ul>
rm	<ul style="list-style-type: none"><li>• Checked if operands missing or not</li><li>• For rm and rm -v, checked if operand a file or not, and it exists or not.</li><li>• For rm -r, checked if the operand exists or not.</li></ul>

**Test case:**

1. For process

ls

mkdir → missing operands

mkdir /root/hi → permission denied

mkdir eg

```
ls
cd eg
ls
cd
ls
rm -r dum1
rm -r eg → file doesn't exist
ls -l
ls -r
ls -p → invalid flag
touch hi1.txt
touch hi2.txt
rm hi1.txt
rm -v h12.txt → file does not exist
rm -v hi2.txt
```

**Assuming h.txt already exists and has some text**

```
cat h.txt
cat gg → file doesn't exist
cat /root → operand not a file, can't use cat on it
cat -n h.txt
cat -b h.txt
mkdir -v dum1
mkdir -p dum1 →doesn't throw error even though dum1 already exists
date
date -u
date -R
date -l → invalid flag
pwd
pwd -L
pwd -P
pwd -f → invalid flag
echo
```

echo -n hello there  
echo hello there  
echo - -help

## 2. For threads

ls &t  
mkdir &t → missing operands  
mkdir /root/hi &t → permission denied  
mkdir eg &t  
ls &t  
cd eg  
ls &t  
cd  
ls &t  
rm -r dum1 &t  
rm -r eg &t → file doesn't exist  
ls -l &t  
ls -r &t  
ls -p &t → invalid flag  
touch hi1.txt &t  
touch hi2.txt &t  
rm hi1.txt &t  
rm -v h12.txt &t → file does not exist  
rm -v hi2.txt &t  
**Assuming h.txt already exists and has some text**  
cat h.txt &t  
cat gg &t → file doesn't exist  
cat /root &t → operand not a file, can't use cat on it  
cat -n h.txt &t  
cat -b h.txt &t  
mkdir -v dum1 &t

mkdir -p dum1 &t →doesn't throw error even though dum1 already exists

date &t

date -u &t

date -R &t

date -l &t → invalid flag

pwd

pwd -L

pwd -P

pwd -f → invalid flag

echo

echo -n hello there

echo hello there

echo - -help