

Playment hiring challenge

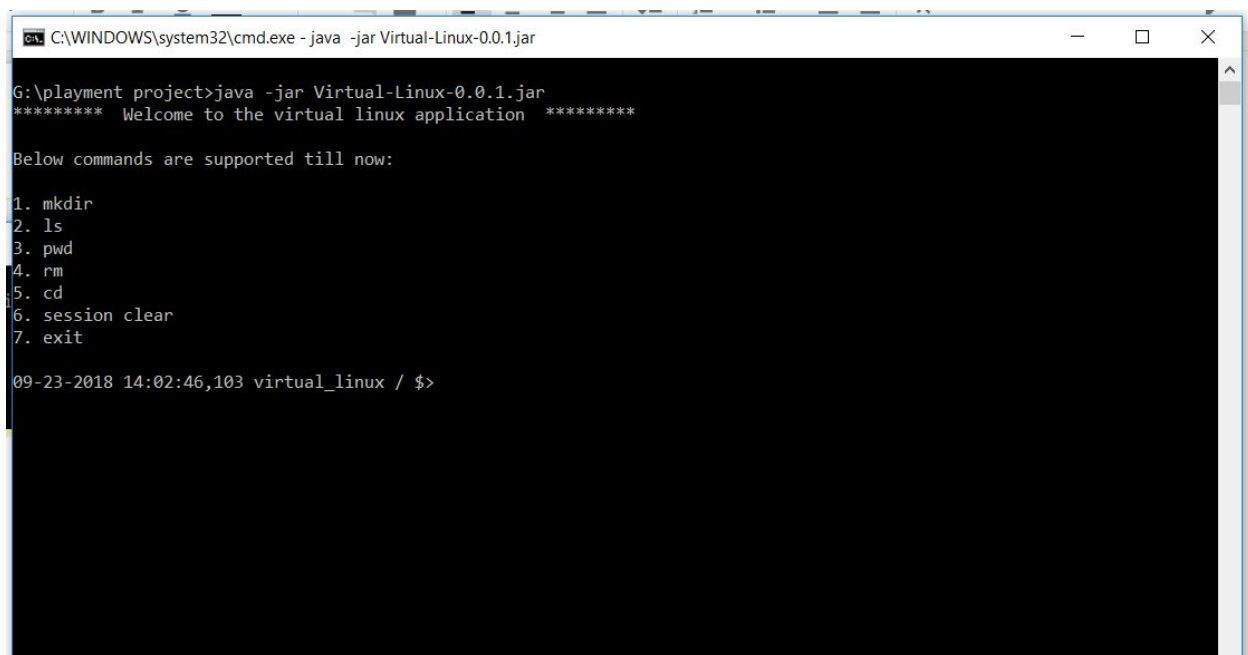
Virtual-Linux Application

Start Application using command 'java -jar Virtual-Linux-0.0.1.jar'. It will display all supported commands. Start using commands one by one.

Assumptions:

1. Command prefixed with / will be executed from root directory
2. Command that are not prefixed with / will be executed on current directory
3. No restrictions for directory name.

Please take a look at the complete document for detailed usage. Each command has been explained separately with image, user can directly read that too.

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe - java -jar Virtual-Linux-0.0.1.jar'. The command prompt shows the user running 'java -jar Virtual-Linux-0.0.1.jar' from the directory 'G:\playment project'. The application outputs a welcome message and a list of supported commands: mkdir, ls, pwd, rm, cd, session clear, and exit. The prompt then shows the date and time '09-23-2018 14:02:46,103' followed by 'virtual_linux / \$>'.

```
C:\WINDOWS\system32\cmd.exe - java -jar Virtual-Linux-0.0.1.jar

G:\playment project>java -jar Virtual-Linux-0.0.1.jar
***** Welcome to the virtual linux application *****

Below commands are supported till now:

1. mkdir
2. ls
3. pwd
4. rm
5. cd
6. session clear
7. exit

09-23-2018 14:02:46,103 virtual_linux / $>
```

mkdir command:

1. mkdir <folder-name> - creates folder with <folder-name> in the **current directory**. You can create n number of folders by simply giving folder names separated by space.
2. mkdir /<folder-name> - creates folder with <folder-name> in the **root directory** because it is prefixed by slash. You can create n number of folders by simply giving folder names separated by space.
3. mkdir <folder-name> /<folder-name> - You can create folders with above combination. It creates respective folders in the corresponding places.

- a. For example: 'mkdir playment hiring success' - this command creates three folders in current directory with the name plyment, hiring and success
 - b. 'mkdir playment /hiring success' - this command creates 'playment' and 'success' in the current directory and 'hiring' in the root directory
4. User can deeply create folders.
 - a. For example: 'mkdir /playment/hiring/success' - this command creates 'playment' directory in the root[prefixed by slash] and 'hiring' directory inside 'playment' and 'success' directory inside the 'hiring' directory.
 - b. 'mkdir playment/hiring/success' creates same as above from current directory.
5. mkdir command is fail safe. It creates inner directory even user inputs already existed directory.
 - a. For example 'mkdir /playment/hiring/success' - Say playment directory already existed, then it only creates hiring directory inside it. It doesn't fail by throwing simply already existed message.
6. This command prints the fully qualified path from root, after successful creation.
7. You can input multiple arguments for an command with combination of above all.

Please take a look at the image below for mkdir command:

```

C:\WINDOWS\system32\cmd.exe - java -jar Virtual-Linux-0.0.1.jar
G:\playment project>java -jar Virtual-Linux-0.0.1.jar
***** Welcome to the virtual linux application *****

Below commands are supported till now:

1. mkdir
2. ls
3. pwd
4. rm
5. cd
6. session clear
7. exit

09-23-2018 14:02:46,103 virtual_linux / $>mkdir playment /challenge/success /hiring/success /virtual_linux
SUCC: CREATED SUCCESSFULLY - FULL PATH: //playment
SUCC: CREATED SUCCESSFULLY - FULL PATH: //challenge
SUCC: CREATED SUCCESSFULLY - FULL PATH: //challenge/success
SUCC: CREATED SUCCESSFULLY - FULL PATH: //hiring
SUCC: CREATED SUCCESSFULLY - FULL PATH: //hiring/success
SUCC: CREATED SUCCESSFULLY - FULL PATH: //virtual_linux

09-23-2018 14:31:00,123 virtual_linux / $>mkdir playment /challenge/success /hiring/success /virtual_linux
ERR: playment ALREADY EXISTED
ERR: ALREADY EXISTED - FULL PATH: //challenge
ERR: ALREADY EXISTED - FULL PATH: //challenge/success
ERR: ALREADY EXISTED - FULL PATH: //hiring
ERR: ALREADY EXISTED - FULL PATH: //hiring/success
ERR: ALREADY EXISTED - FULL PATH: //virtual_linux

09-23-2018 14:37:34,030 virtual_linux / $>
  
```

ls command:

1. This command lists all the folders in the current directory.
2. If there is no directory exists, then it will print 'NO DIRECTORY EXISTS' message.

Please take a look at the image below for ls command:

```
09-23-2018 14:37:34,030 virtual_linux / $>ls
DIRS: playment challenge hiring virtual_linux
```

```
09-23-2018 14:42:35,091 virtual_linux //playment $>ls
DIRS: NO DIRECTORY EXIST
```

pwd command:

1. It prints current directory. It has no arguments.

Please take a look at the image below for pwd command:

```
09-23-2018 14:46:43,935 virtual_linux / $>cd challenge
SUCC: REACHED: //challenge

09-23-2018 14:46:51,060 virtual_linux //challenge $>ls
DIRS: success

09-23-2018 14:46:52,676 virtual_linux //challenge $>cd success
SUCC: REACHED: //challenge/success

09-23-2018 14:46:58,216 virtual_linux //challenge/success $>pwd
PATH: //challenge/success
```

rm command:

1. This command removes the given directory.
2. `rm <folder-name>` - removes folder if exists in **current directory**.
 - a. For example: `rm playment` - removes folder if it is in corresponding directory or else throw error message as not existed
3. `rm /<folder-name>` - removes folder if exists in **root directory**. It also checks whether given folder is current directory or parent of current directory. If so, It doesn't removes and throws error message.
 - a. For example: Let's assume folder structure as 'playment/hiring/success' and you are inside success directory. At this point user enters '`rm /playment`' command. It will show an error because you cannot remove current directory or its parent.
4. User can remove multiple folders at a time.
 - a. For example: '`rm hiring /challenge fail`' - It removes 'hiring' and 'fail' in the current directory and removes 'challenge' in root directory.
5. It supports deep delete like '`rm <folder-name>/<folder-name>`'.
 - a. For example - '`rm play/fail`', Let's assume current directory as 'chal' then it removes fail directory from the current directory(deleted path: chai/play/fail).
 - b. '`rm /play/fail`' - Let's assume current directory as 'chal' then it removes fail directory from the root directory(deleted path: /play/fail).

- c. Above both command executes only if it executes in corresponding directory, otherwise Invalid path message thrown

Please take a look at the below image for rm command:

```
09-23-2018 15:01:08,662 virtual_linux //playment $>rm /playment /challenge challenge
ERR: CANNOT REMOVE CURRENT DIRECTORY OR ITS PARENT
SUCC: DELETED
ERR: DIRECTORY DOESN'T EXIST
```

cd command:

1. cd command navigates to the given directory. It has only one arguments.
2. cd <folder-name> - navigates to the corresponding folder
3. cd / - navigates to the root folder
4. cd <folder-name>/<folder-name>/<folder-name> - navigates to the corresponding path from **current directory**.
5. cd /<folder-name>/<folder-name>/<folder-name> - navigates to the corresponding path from **root directory**[prefixed by /].
6. If path is invalid, it doesn't navigates and throws error message.

Please take a look at the below image for cd command:

```
09-23-2018 15:16:21,391 virtual_linux / $>mkdir playment/hiring/challenge
SUCC: CREATED SUCCESSFULLY - FULL PATH: //playment
SUCC: CREATED SUCCESSFULLY - FULL PATH: //playment/hiring
SUCC: CREATED SUCCESSFULLY - FULL PATH: //playment/hiring/challenge

09-23-2018 15:16:25,639 virtual_linux / $>cd /playment/hiring
SUCC: REACHED: //playment
SUCC: REACHED: //playment/hiring

09-23-2018 15:16:45,483 virtual_linux //playment/hiring $>pwd
PATH: //playment/hiring

09-23-2018 15:16:56,308 virtual_linux //playment/hiring $>cd challenge
SUCC: REACHED: //playment/hiring/challenge

09-23-2018 15:17:08,509 virtual_linux //playment/hiring/challenge $>pwd
PATH: //playment/hiring/challenge

09-23-2018 15:17:11,464 virtual_linux //playment/hiring/challenge $>cd /
SUCC: REACHED TO ROOT DIRECTORY

09-23-2018 15:17:14,825 virtual_linux / $>pwd
PATH: /
```

* It also prints current directory before the dollar symbol.

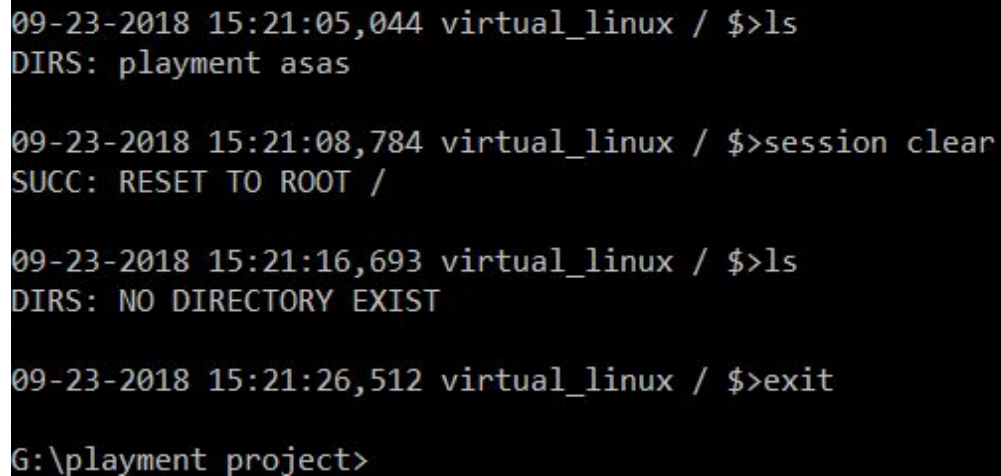
session clear command:

1. This command clears all data and reset to root.
2. Session command can only supports clear as arguments now.

exit command:

1. Exits application smoothly.

Please take a look at the image for session and exit command:

A terminal window with a black background and green text. It shows four lines of command execution. The first line shows a timestamp '09-23-2018 15:21:05,044', the prompt 'virtual_linux / \$', and the command 'ls', followed by the output 'DIRS: playment asas'. The second line shows a timestamp '09-23-2018 15:21:08,784', the prompt 'virtual_linux / \$', and the command 'session clear', followed by the output 'SUCC: RESET TO ROOT /'. The third line shows a timestamp '09-23-2018 15:21:16,693', the prompt 'virtual_linux / \$', and the command 'ls', followed by the output 'DIRS: NO DIRECTORY EXIST'. The fourth line shows a timestamp '09-23-2018 15:21:26,512', the prompt 'virtual_linux / \$', and the command 'exit'. The final line shows the Windows command prompt 'G:\playment project>'.

```
09-23-2018 15:21:05,044 virtual_linux / $>ls
DIRS: playment asas

09-23-2018 15:21:08,784 virtual_linux / $>session clear
SUCC: RESET TO ROOT /

09-23-2018 15:21:16,693 virtual_linux / $>ls
DIRS: NO DIRECTORY EXIST

09-23-2018 15:21:26,512 virtual_linux / $>exit

G:\playment project>
```

Regards,
Vignesh Baskaran.