

USER MANUAL

GROUP : 16

SLOT : 11

STAGE 2 SUBMISSION

Project – Unix Shell in C++ with some mathematical functions

Project Name – MaSH

Input :

User can input simple *nix commands. User will input command at the prompt. Inputting illegal will result in shell spitting out error in execvp.

Example

```
user@system:~/home/user/Documents/project CS101/mash_test # ls
headers.h  headers.h~  mash  small_shell.cpp  small_shell.cpp~
utility.cpp  utility.cpp~

user@system:~/home/user/Documents/project CS101/mash_test # cd ..

user@system:~/home/user/Documents/project CS101 # ls
bin      mash      mash_22  mash_test  old_mash      small_shell.cpp
SRS.doc      test

headers.h~  mash_21  mash.cpp~  mini-shell  project  report.doc
small_shell.cpp~  stage 1  upload  utility.cpp~

user@system:~/home/user/Documents/project CS101 # cal

      October 2014
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

user@system:~/home/user/Documents/project CS101 #
```

This is an example for running MaSH. Complex commands such as piping and I/O redirection has still not been implemented.

For mathematics we are using `fparser v4.4.3` for C++.

NOTE: Function commands are CASE SENSITIVE.

1. a^b - a raised to the power b
2. $-a$ - unary minus

3. $a*b$ a/b $a\%b$ – multiplication, division, modulo
4. $a + b$, $a - b$ – addition and subtraction
5. $\text{abs}(A)$, $\text{floor}(a)$ – Absolute value of a , box of a
6. $\text{cos}(a)$, $\text{sin}(a)$, $\text{tan}(a)$, $\text{cot}(a)$, $\text{csc}(a)$, $\text{sec}(a)$ – Trigonometric functions
7. $\text{cbrt}(a)$, $\text{sqrt}(a)$ – Cuberoot and squareroot of a
8. $\text{exp}(a)$, $\text{log}(a)$ – Exponential and logarithm with base e of a
9. $\text{exp2}(a)$, $\text{log2}(a)$ – Exponential and logarithm with base e of a

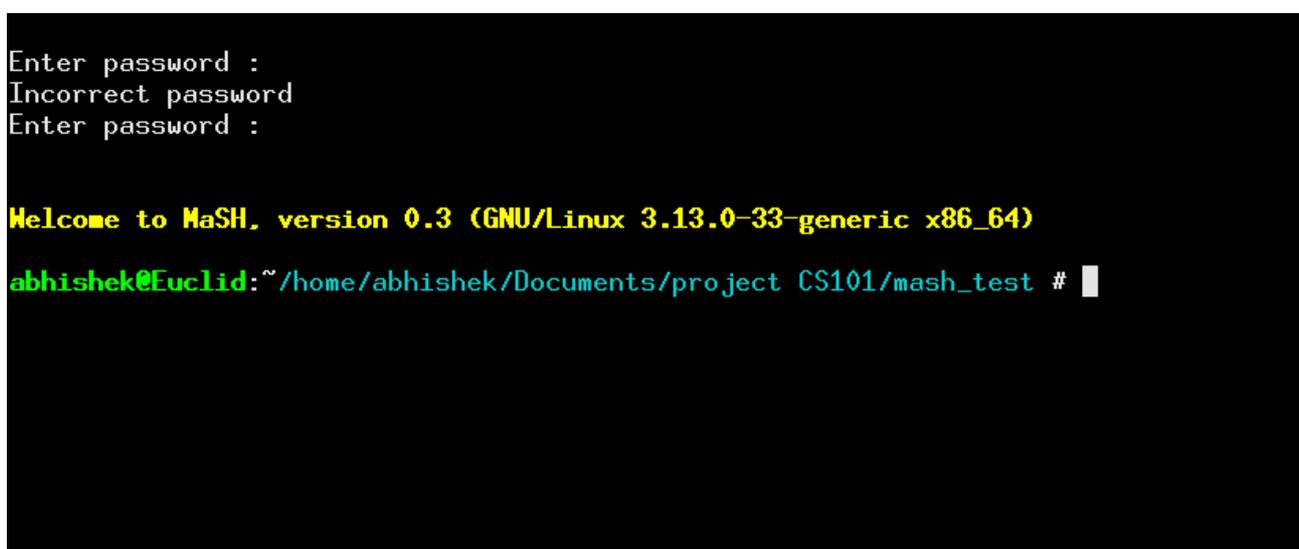
STEP 1 :

The shell will ask for password. For the first time password will be provided in the package in a file named shadow.txt.



STEP 2 :

After entering the password, shell will greet you with a welcome screen. From here on, you can carry on with you daily Shell command working.



MaSh in action :

```
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test # ls
a.out      logs.txt  mashv0.1  matrix2.cpp  matrix.cpp  password.cpp  run.sh~      test_exec.cpp~
bin        main.cpp  mashv0.2  matrix2.cpp~  matrix.cpp~  rdp.o        small_shell.cpp~  utility.cpp
headers.h  main.cpp~  mashv0.3  matrix3.cpp  matrix.o    recursive_descent_parser.cpp  system3.cpp    utility.cpp~
headers.h~  mash      mashv0.3.o  matrix3.cpp~  matrix.o    recursive_descent_parser.cpp~  test_exec.cpp

abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test # cd ..
abhishek@Euclid:~/home/abhishek/Documents/project CS101 # ls
11_16_MaSH.tar.gz  fparser      Makefile~  mash_22      mash_test    old_mash      SRS.doc      user_manual.doc
bin                headers.h~    mash       mash.cpp~    mash updated  project report.doc  stage 1 upload  utility.cpp~
findroots..cpp    integration.cpp  mash_21    mash_final   mashv1       small_shell.cpp~  test

abhishek@Euclid:~/home/abhishek/Documents/project CS101 # mkdir HELLO
abhishek@Euclid:~/home/abhishek/Documents/project CS101 # cd HELLO
abhishek@Euclid:~/home/abhishek/Documents/project CS101/HELLO # ls $HOME
a.out
b.c
b.cpp
b.cpp~
Beginner Drumming Independence - Free Drum Lessons-t0G-jcQcwmc.mp4
c.c
Desktop
Documents
Downloads
examples.desktop
marks_statement.pdf
Music
peazip_portable-5.3.1.LINUX.Qt.tar.gz
Pictures
Public
Rock Drum Fills - Free Beginner Drum Lessons (Part #1 of 5)-06Yd7XnYcPU.mp4
Templates
Videos
abhishek@Euclid:~/home/abhishek/Documents/project CS101/HELLO # jkshd
Error 'execvp()': : No such file or directory
abhishek@Euclid:~/home/abhishek/Documents/project CS101/HELLO # █
```

MaSh supports environmental variables like \$HOME. As it can be seen in the picture above that `ls $HOME` lists all the file in user home directory.

```
abhishek@Euclid:~/home/abhishek/Documents/project CS101/HELLO # cd ..
abhishek@Euclid:~/home/abhishek/Documents/project CS101 # cd mash_test
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test # cd bin
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # ls
differentiate.cpp  docs      eval.o    findroots.cpp  fparser_gmpint.hh  fpconfig.hh  gcd.cpp  integrate.cpp  matrix.o
differentiate.cpp~  eval.cpp  examples  findroots.cpp~  fparser.hh        fpoptimizer.cc  gcd.cpp~  integrate.cpp~  mpfr
differentiate.o    eval.cpp~  extrasrc  fparser.cc     fparser_mpfr.hh   froot.o       gcd.o    integrate.o

abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # ls | grep cpp
differentiate.cpp
differentiate.cpp~
eval.cpp
eval.cpp~
findroots.cpp
findroots.cpp~
gcd.cpp
gcd.cpp~
integrate.cpp
integrate.cpp~
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # █
```

Here we can see that MaSh supports two process piping. It is demonstrated by the command `ls | gerp cpp` which lists all the files having `cpp` in their name.

MaSH also supports builtin command history, which prints all the commands previously used in MaSH, even the if MaSH is exited in between.

```
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # history
history
version
fgfhgf
math ghggg
math
math
math eval
math eval
math matrix
clear
ls
cd ..
ls
mkdir HELLO
cd HELLO
ls $HOME
jkshd
clear
cd ..
cd mash_test
cd bin
ls
ls | grep cpp
ls | grep cpp >> a.txt
clear
history
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin #
```

History can be cleared by the command `history -c`

```
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # history -c
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # history
history
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin #
```

General mathematics operations in MaSH. Here math is a builtin command in MaSH.

```
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math
Error : No command given for Math
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math hgdh
Error : Command not found
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math eval
f(x) = sin(x)
Point at which value has to be evaluated : 0
f(0) = 0
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math integrate
f(x) = x^2
min x: 0
max x: 1
integration of f(x) is 0.333333
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math froot
f(x) = x^2 - 4
Enter an approximate root : -4
One of the root of f(x) : -2
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math differentiate
f(x) = x^2 + 1
Enter point : 2
Value of the f'(x) at 2 : 4
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math gcd
Enter the number of number whose GCD has to be found : 4
Enter the numbers :
8 4 16 24
GCD : 4
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin #
```

Matrix manipulation in MaSH. It must be noted that when rows and columns of A and B are prompted they must be entered as follow a_row >> a_column >> b_row >> b_column

```
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin # math matrix
Menu driven Matrix manipulation program :

1. Add or Subtract Matrix
2. Multiply two Matrices
3. Find determinant (upto 3x3)
4. Exit
:1

Enter rows and columns for matrix A and B respectively :
2 2 2 2

Enter matrix A :
1 2
0 1

Enter matrix B :
2 1
3 1

For addition or subtraction press '+' or '-' : +
3 3
3 2

1. Add or Subtract Matrix
2. Multiply two Matrices
3. Find determinant (upto 3x3)
4. Exit
:4
abhishek@Euclid:~/home/abhishek/Documents/project CS101/mash_test/bin #
```

Builtin Command `exit` must be used for logging out, as depicted below.

```
abhishek@Euclid:~/home/abhishek/Documents/project_CS101/mash_test/bin # cd /
abhishek@Euclid:~/ # exit
Logging out
abhishek@Euclid:~/Documents/project_CS101/mash_test$ █
```

Thank You