

CL205 - Operating Systems Lab

Task#02

Q1. Write a program in C that imitates the shell. Your program should be able to run at least 3 shell commands. Name the commands so that they are meaningful e.g. Use `showprocesses` instead of `ps`. The user will pass the command as a parameter to the program.

Sample Output

```
jack@PC1:~$ ./com showprocesses
```

PID	TTY	TIME	CMD
3051	pts/0	00:00:00	bash
3131	pts/0	00:00:00	ps

Where *com* is the output file of the C program and *showprocesses* is the command passed to the program as a parameter.

Q2. Make changes to the program in Q1 so that whenever the program is being run the user is prompted for a command instead of passing it as a parameter to the program. The program should execute the command and then prompt for another command, the program should keep running until the user types *exit*.

Sample Output

```
jack@PC1:~$ ./com
```

Myshell#:			
arg	arg.c	com	com.c
exec	exec.c	first	first.c
processes			

```
Myshell#:
```

PID	TTY	TIME	CMD
3051	pts/0	00:00:00	bash
4079	pts/0	00:00:00	com
4081	pts/0	00:00:00	ps

```
Myshell#:
```

```
exit
```

```
jack@PC1:~$
```

Q3. Make changes to the program in Q2 so that the user can use parameters with the shell commands.

Sample Output

```
jack@PC1:~$ ./com
Myshell#:~ list
arg arg.c com com.c exec exec.c first first.c processes
Myshell#:~ showprocesses indetail
USER  PID  %CPU  %MEM  VSZ   RSS  TTY   STAT  START  TIME  COMMAND
root   1180   0.0   0.0  34144  1828  tty1  Ss+   23:28   0:00  /sbin/agetty --noclear tty1 linux
jack   3051   0.0   0.1  40760  5088  pts/0  Ss    23:37   0:00  /bin/bash
jack   4166   0.0   0.0  55568  3392  pts/0  R+    23:58   0:00  ps au
Myshell#:~ exit
jack@P-C1:~$
```