*LAB # 03*

Fprocess MAnagement

# *OBJECTIVE:*

*Implementation of process management using the following system calls of LINUX operating system: fork, getpid, getppid.*

*LAB task*

***Program01***

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  int main()  {  int pid;  pid = getpid( );  printf("Process ID is %d \n",pid);  } |

***Output:***

Text

Description automatically generated

***Program02***

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  int main()  {  int ppid;  ppid = getppid( );  printf("Parent Process ID is %d\n",ppid);  } |

Text

Description automatically generated***Program03***

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  int main()  {  fork( );  printf("Hello World\n");  } |

***Output:***

Text

Description automatically generated

**Program04**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  int main()  {  int pid;  pid = fork( );  if(pid>0)  printf("Parent process PID is %d\n",pid);  } |

***Output:***

Text

Description automatically generated***Program05***

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  int main()  {  int pid;  pid = fork( );  if (pid == 0)  printf("Child Process\n");  } |

***Output***

Text

Description automatically generated

***Program06***

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  int main()  {  int pid;  pid=fork( );  if(pid==0)  {  printf("I am the child, my process ID is %d\n",getpid( ));  printf("The child’s parent process ID is %d\n",getppid( ));  }  else  {  printf("I am the parent, my process ID is %d\n",getpid( ));  printf("The parents parent process ID is %d\n",getppid( ));  } } |

Text, letter

Description automatically generated

***Output:***

## *Write a program to get Process Id and Parent Process Id in one program also attach the screenshot of output.*

|  |
| --- |
| #include <stdio.h>#include <unistd.h>int main() {int pid;int ppid;pid = getpid(); /\*process id\*/ppid = getpid(); /\*parent process id\*/printf("Process ID: %d\n", pid);printf("Parent Process ID: %d\n", ppid);} |

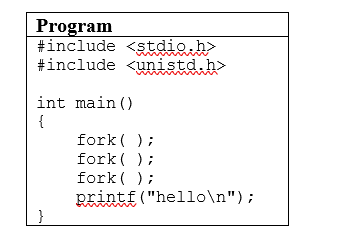
***Output:***

Text

Description automatically generated

1. *Show the output of the given code and justify in your own words that why this output is occurring?*

***Code:***

******

***Output:***

Text

Description automatically generated

***Justification:***

*Fork() works for both parent class & child class, after executing child class… the fork() will move to parent class & will execute it. Hence, hello is printed 6 times, one time for child class & one time for parent class.*

*Eg; fork() 🡪child class*

*🡪parent class*

*fork() 🡪child class*

*🡪parent class*

*fork() 🡪child class*

*🡪parent class*