Write a program to create child process using fork() command.

**Description:**

Fork system call is used for creating a new process, which is called ***child process***, which runs concurrently with the process that makes the fork() call (parent process).

After a new child process is created, both processes will execute the next instruction following the fork() system call. A child process uses the same pc(program counter), same CPU registers, same open files which use in the parent process.

It takes no parameters and returns an integer value.

Below are different values returned by fork().

***Negative Value***: creation of a child process was unsuccessful.  
***Zero***: Returned to the newly created child process.  
***Positive value***: Returned to parent or caller. The value contains process ID of newly created child process.

**Program:**

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

// make two process which run same

fork();

printf("Hello world!\n");

return 0;

}

**Output:**

$ ./a.out

Hello world!

Hello world!

Write a program to print child and parent process using fork() command.

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

void forkexample()

{

// child process because return value zero

if (fork() == 0)

{

printf("Hello from Child!\n");

}

// parent process because return value non-zero.

else

printf("Hello from Parent!\n");

}

int main()

{

forkexample();

return 0;

}

**Output:**

./a.out

Hello from Parent!

Hello from Child!