// hello.c

#include <stdio.h>

int main(int argc, char \*argv[])

{

if (argc != 2)

{

fprintf(stderr, "Usage: %s <name>\n", argv[0]);

return 1;

}

printf("Hello, %s!\n", argv[1]);

getch ();

return 0;

}

// Child process creation

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/wait.h>

int main()

{

pid\_t pid;

// Create a child process

pid = fork();

if (pid < 0)

{

fprintf(stderr, "Fork failed.\n");

return 1;

}

else if (pid == 0)

{

// This code is executed by the child process

printf("Child process executing hello program:\n");

execlp("./hello", "hello", "John", NULL); // Execute the hello program with the name "John"

}

else

{

// This code is executed by the parent process

printf("Parent process waiting for the child to finish...\n");

wait(NULL); // Wait for the child process to finish

printf("Parent process finished.\n");

}

return 0;

}

: gcc hello.c -o hello

: gcc main.c -o main

: ./main

Output:

Parent process waiting for the child to finish...

Child process executing hello program:

Hello, John!

Parent process finished.