Write another program using fork(). The child process should print “hello”; the parent process should print “goodbye”. You should try to ensure that the child process always prints first; can you do this without calling wait() in the parent?

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <unistd.h>*

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

{

int myVariable = 100;

printf("hai world (pid:%d)\n", (int)getpid());

int rc = fork();

if (rc < 0)

{

*// fork failed;*

fprintf(stderr, "fork failed\n");

exit(1);

}

else if (rc == 0)

{

*// child (new process)*

printf("hello, I am child (pid:%d)\n", (int)getpid());

}

else

{

for (int i = 0; i < 10000; i++)

{

;

}

*// parent goes down this path (original process)*

printf("goodbye, I am parent of %d (pid:%d)\n",

rc, (int)getpid());

}

return 0;

}

**Output:**

hai world (pid:994)

hello, I am child (pid:995)

goodbye, I am parent of 995 (pid:994)