MakeCopy.c : Write a C program that makes a new copy of an existing file using system calls for file manipulation. The names of the two files and copy block sizes are to be specified as command line arguments. Open the source file in read only mode and destination file in read/write mode.

```
FILE *src;

src  = fopen(argv[1], "r");

if(src == NULL) {

   printf("Error: Could NOT open file \"%s\".\n", argv[1]);

   exit(-1);

}


FILE *dest;

dest = fopen(argv[2], "w+");

if(dest == NULL) {

   printf("Error: Could NOT open file \"%s\".\n", argv[2]);

   fclose(src);

   exit(-1);

}


char *buffer[bufferSize];

while (1) {

   int read_chars = fread(buffer, sizeof(char), bufferSize-1, src);

   if(read_chars <= 0) break;

   fwrite(buffer, sizeof(char), read_chars, dest);

}


fclose(src);

fclose(dest);
```

ForkCopy.c : Write a C program that creates a new process to copy the files using the MakeCopy. This program should spawn a new process using fork system call. Then use execl to execute MakeCopy program. The source and destination file names presented as command-line arguments should be passed to execl as system call arguments. The main process waits for completion of copy operation using wait system call.

```
pid_t child_pid;

int child_status;

   child_pid = fork();

   switch (child_pid) {

      case -1:

         perror("fork");

         exit(1);

      case 0:

         execl("./MakeCopy", "MakeCopy", argv[1], argv[2], argv[3], NULL);

         perror("execl() failed!");

      default:

         wait(&child_status);

   }
```