

Doug Branton
COSC519
Homework 2

1. Modify the hello.c program to open an input file (input.txt), read from the input file, and write to another output file (output.txt). This program reads text from one file and writes to another file. Create some text data in the input file and verify that the same data is written to the output file. Understand how a system call is invoked and how it works by generating and reading an ASM file. Identify and mark the system calls in your ASM file. Submit your hello.c and ASM files showing the system calls (Use Linux).

Hello.c:

```
//This is first program
//Dr. Karne
//hello.c
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char **argv)
{
    char c1;
    unsigned char c2;
    int i1=0;
    long l2=0;
    char *cptr;
    int *iptr;
    long *lptr;
    char array1[40] = "This is a string";

    cptr = (char *)malloc(200);
    iptr = (int *)malloc(200);
    lptr = (long *)malloc(200);

    c1 = 'X';
    c2 = 0x44;
    i1 = 0x100;
    l2 = 0x0123456789abcdef;

    *iptr = 0x2000;
    *lptr = 0x88889999aaaabbbb;

    printf("Hello World\n");
    printf("\n\n");
    printf("l2: %lx \n", l2);
    printf("i1: %x \n", i1);
    printf("i1: %10x \n", i1);
    printf("i1: %4x \n", i1);
    printf("c1: %c \n", c1);
```

```

printf("string: %s \n", array1);

/*Copy from input.txt to output.txt */
FILE *input = fopen("input.txt","r");
FILE *output = fopen("output.txt", "w");

char ch;

while((ch = fgetc(input)) != EOF)
    fputc(ch, output);

fclose(input);
fclose(output);

return 0;
}

```

Hello.s (ASM File):

```

.file    "hello.c"
.intel_syntax noprefix
.text
.Ltext0:
.section     .rodata
.LC0:
.string "Hello World"
.LC1:
.string "\n"
.LC2:
.string "l2: %lx \n"
.LC3:
.string "i1: %x \n"
.LC4:
.string "i1: %10x \n"
.LC5:
.string "i1: %4x \n"
.LC6:
.string "c1: %c \n"
.LC7:
.string "string: %s \n"
.LC8:
.string "r"
.LC9:
.string "input.txt"
.LC10:
.string "w"
.LC11:
.string "output.txt"

```

```

.text
.globl main
.type main, @function
main:
.LFB6:
.file 1 "hello.c"
.loc 1 8 1
.cfi_startproc
endbr64
push rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
mov rbp, rsp
.cfi_def_cfa_register 6
add rsp, -128
mov DWORD PTR -116[rbp], edi
mov QWORD PTR -128[rbp], rsi
.loc 1 8 1
mov rax, QWORD PTR fs:40
mov QWORD PTR -8[rbp], rax
xor eax, eax
.loc 1 11 8
mov DWORD PTR -100[rbp], 0
.loc 1 12 9
mov QWORD PTR -96[rbp], 0
.loc 1 16 9
movabs rax, 2338328219631577172
movabs rdx, 7453010373645639777
mov QWORD PTR -48[rbp], rax
mov QWORD PTR -40[rbp], rdx
mov QWORD PTR -32[rbp], 0
mov QWORD PTR -24[rbp], 0
mov QWORD PTR -16[rbp], 0
.loc 1 18 19
mov edi, 200
call malloc@PLT
mov QWORD PTR -88[rbp], rax
.loc 1 19 18
mov edi, 200
call malloc@PLT
mov QWORD PTR -80[rbp], rax
.loc 1 20 19
mov edi, 200
call malloc@PLT
mov QWORD PTR -72[rbp], rax
.loc 1 22 7
mov BYTE PTR -103[rbp], 88
.loc 1 23 7
mov BYTE PTR -102[rbp], 68

```

```
.loc 1 24 7
mov     DWORD PTR -100[rbp], 256
.loc 1 25 7
movabs     rax, 81985529216486895
mov     QWORD PTR -96[rbp], rax
.loc 1 27 10
mov     rax, QWORD PTR -80[rbp]
mov     DWORD PTR [rax], 8192
.loc 1 28 10
mov     rax, QWORD PTR -72[rbp]
movabs     rcx, -8608461802446341189
mov     QWORD PTR [rax], rcx
.loc 1 30 4
lea     rdi, .LC0[rip]
call    puts@PLT
.loc 1 31 4
lea     rdi, .LC1[rip]
call    puts@PLT
.loc 1 32 4
mov     rax, QWORD PTR -96[rbp]
mov     rsi, rax
lea     rdi, .LC2[rip]
mov     eax, 0
call    printf@PLT
.loc 1 33 4
mov     eax, DWORD PTR -100[rbp]
mov     esi, eax
lea     rdi, .LC3[rip]
mov     eax, 0
call    printf@PLT
.loc 1 34 4
mov     eax, DWORD PTR -100[rbp]
mov     esi, eax
lea     rdi, .LC4[rip]
mov     eax, 0
call    printf@PLT
.loc 1 35 4
mov     eax, DWORD PTR -100[rbp]
mov     esi, eax
lea     rdi, .LC5[rip]
mov     eax, 0
call    printf@PLT
.loc 1 36 4
movsx   eax, BYTE PTR -103[rbp]
mov     esi, eax
lea     rdi, .LC6[rip]
mov     eax, 0
call    printf@PLT
.loc 1 37 4
```

```

lea    rax, -48[rbp]
mov    rsi, rax
lea    rdi, .LC7[rip]
mov    eax, 0
call   printf@PLT
.loc 1 40 18
lea    rsi, .LC8[rip]
lea    rdi, .LC9[rip]
call   fopen@PLT
mov    QWORD PTR -64[rbp], rax
.loc 1 41 19
lea    rsi, .LC10[rip]
lea    rdi, .LC11[rip]
call   fopen@PLT
mov    QWORD PTR -56[rbp], rax
.loc 1 45 9
jmp    .L2

```

.L3:

```

.loc 1 46 7
movsx  eax, BYTE PTR -101[rbp]
mov    rdx, QWORD PTR -56[rbp]
mov    rsi, rdx
mov    edi, eax
call   fputc@PLT

```

.L2:

```

.loc 1 45 16
mov    rax, QWORD PTR -64[rbp]
mov    rdi, rax
call   fgetc@PLT
.loc 1 45 14
mov    BYTE PTR -101[rbp], al
.loc 1 45 9
cmp    BYTE PTR -101[rbp], -1
jne    .L3
.loc 1 48 4
mov    rax, QWORD PTR -64[rbp]
mov    rdi, rax
call   fclose@PLT
.loc 1 49 4
mov    rax, QWORD PTR -56[rbp]
mov    rdi, rax
call   fclose@PLT
.loc 1 52 11
mov    eax, 0
.loc 1 53 1
mov    rcx, QWORD PTR -8[rbp]
xor    rcx, QWORD PTR fs:40
je     .L5
call   __stack_chk_fail@PLT

```

```

.L5:
    leave
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE6:
    .size    main, .-main
.Letext0:
    .file 2 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h"
    .file 3 "/usr/include/x86_64-linux-gnu/bits/types.h"
    .file 4 "/usr/include/x86_64-linux-gnu/bits/types/struct_FILE.h"
    .file 5 "/usr/include/x86_64-linux-gnu/bits/types/FILE.h"
    .file 6 "/usr/include/stdio.h"
    .file 7 "/usr/include/x86_64-linux-gnu/bits/sys_errlist.h"
    .section      .debug_info,"",@progbits
.Ldebug_info0:
[OMITTED DEBUG INFO FOR LENGTH]

```

2. Using the above hello.exe or hello.o files, run objdump command to find system calls and mark them in a file. System calls have UND symbols.

hello: file format elf64-x86-64

Disassembly of section .init:

```

0000000000001000 <_init>:
1000:    f3 0f 1e fa          endbr64
1004:    48 83 ec 08          sub    $0x8,%rsp
1008:    48 8b 05 d9 2f 00 00  mov    0x2fd9(%rip),%rax    # 3fe8 <__gmon_start__>
100f:    48 85 c0              test   %rax,%rax
1012:    74 02                je     1016 <_init+0x16>
1014:    ff d0                callq  *%rax
1016:    48 83 c4 08          add    $0x8,%rsp
101a:    c3                  retq

```

Disassembly of section .plt:

```

0000000000001020 <.plt>:
1020:    ff 35 62 2f 00 00    pushq 0x2f62(%rip)    # 3f88
<_GLOBAL_OFFSET_TABLE_+0x8>
1026:    f2 ff 25 63 2f 00 00  bnd jmpq *0x2f63(%rip)    # 3f90
<_GLOBAL_OFFSET_TABLE_+0x10>
102d:    0f 1f 00              nopl   (%rax)
1030:    f3 0f 1e fa          endbr64
1034:    68 00 00 00 00        pushq $0x0
1039:    f2 e9 e1 ff ff ff    bnd jmpq 1020 <.plt>
103f:    90                    nop
1040:    f3 0f 1e fa          endbr64

```

```

1044: 68 01 00 00 00    pushq $0x1
1049: f2 e9 d1 ff ff ff bnd jmpq 1020 <.plt>
104f: 90                nop
1050: f3 0f 1e fa       endbr64
1054: 68 02 00 00 00    pushq $0x2
1059: f2 e9 c1 ff ff ff bnd jmpq 1020 <.plt>
105f: 90                nop
1060: f3 0f 1e fa       endbr64
1064: 68 03 00 00 00    pushq $0x3
1069: f2 e9 b1 ff ff ff bnd jmpq 1020 <.plt>
106f: 90                nop
1070: f3 0f 1e fa       endbr64
1074: 68 04 00 00 00    pushq $0x4
1079: f2 e9 a1 ff ff ff bnd jmpq 1020 <.plt>
107f: 90                nop
1080: f3 0f 1e fa       endbr64
1084: 68 05 00 00 00    pushq $0x5
1089: f2 e9 91 ff ff ff bnd jmpq 1020 <.plt>
108f: 90                nop
1090: f3 0f 1e fa       endbr64
1094: 68 06 00 00 00    pushq $0x6
1099: f2 e9 81 ff ff ff bnd jmpq 1020 <.plt>
109f: 90                nop
10a0: f3 0f 1e fa       endbr64
10a4: 68 07 00 00 00    pushq $0x7
10a9: f2 e9 71 ff ff ff bnd jmpq 1020 <.plt>
10af: 90                nop

```

Disassembly of section .plt.got:

```

000000000000010b0 <__cxa_finalize@plt>:
 10b0: f3 0f 1e fa       endbr64
 10b4: f2 ff 25 3d 2f 00 00 bnd jmpq *0x2f3d(%rip)    # 3ff8
<__cxa_finalize@GLIBC_2.2.5>
 10bb: 0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

```

Disassembly of section .plt.sec:

```

000000000000010c0 <puts@plt>:
 10c0: f3 0f 1e fa       endbr64
 10c4: f2 ff 25 cd 2e 00 00 bnd jmpq *0x2ecd(%rip)    # 3f98 <puts@GLIBC_2.2.5>
 10cb: 0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

000000000000010d0 <fclose@plt>:
 10d0: f3 0f 1e fa       endbr64
 10d4: f2 ff 25 c5 2e 00 00 bnd jmpq *0x2ec5(%rip)    # 3fa0 <fclose@GLIBC_2.2.5>
 10db: 0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

000000000000010e0 <__stack_chk_fail@plt>:

```

```

10e0:    f3 0f 1e fa      endbr64
10e4:    f2 ff 25 bd 2e 00 00 bnd jmpq *0x2ebd(%rip)    # 3fa8
<__stack_chk_fail@GLIBC_2.4>
10eb:    0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

00000000000010f0 <printf@plt>:
10f0:    f3 0f 1e fa      endbr64
10f4:    f2 ff 25 b5 2e 00 00 bnd jmpq *0x2eb5(%rip)    # 3fb0 <printf@GLIBC_2.2.5>
10fb:    0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

0000000000001100 <fgetc@plt>:
1100:    f3 0f 1e fa      endbr64
1104:    f2 ff 25 ad 2e 00 00 bnd jmpq *0x2ead(%rip)    # 3fb8 <fgetc@GLIBC_2.2.5>
110b:    0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

0000000000001110 <fputc@plt>:
1110:    f3 0f 1e fa      endbr64
1114:    f2 ff 25 a5 2e 00 00 bnd jmpq *0x2ea5(%rip)    # 3fc0 <fputc@GLIBC_2.2.5>
111b:    0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

0000000000001120 <malloc@plt>:
1120:    f3 0f 1e fa      endbr64
1124:    f2 ff 25 9d 2e 00 00 bnd jmpq *0x2e9d(%rip)    # 3fc8 <malloc@GLIBC_2.2.5>
112b:    0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

0000000000001130 <fopen@plt>:
1130:    f3 0f 1e fa      endbr64
1134:    f2 ff 25 95 2e 00 00 bnd jmpq *0x2e95(%rip)    # 3fd0 <fopen@GLIBC_2.2.5>
113b:    0f 1f 44 00 00    nopl 0x0(%rax,%rax,1)

```

Disassembly of section .text:

```

0000000000001140 <_start>:
1140:    f3 0f 1e fa      endbr64
1144:    31 ed            xor  %ebp,%ebp
1146:    49 89 d1          mov  %rdx,%r9
1149:    5e                pop  %rsi
114a:    48 89 e2          mov  %rsp,%rdx
114d:    48 83 e4 f0       and  $0xfffffffffff0,%rsp
1151:    50                push %rax
1152:    54                push %rsp
1153:    4c 8d 05 36 03 00 00 lea  0x336(%rip),%r8    # 1490 <__libc_csu_fini>
115a:    48 8d 0d bf 02 00 00 lea  0x2bf(%rip),%rcx    # 1420 <__libc_csu_init>
1161:    48 8d 3d c1 00 00 00 lea  0xc1(%rip),%rdi    # 1229 <main>
1168:    ff 15 72 2e 00 00 callq *0x2e72(%rip)    # 3fe0
<__libc_start_main@GLIBC_2.2.5>
116e:    f4                hlt
116f:    90                nop

```


0000000000001170 <deregister_tm_clones>:

```
1170: 48 8d 3d 99 2e 00 00 lea 0x2e99(%rip),%rdi    # 4010 <__TMC_END__>
1177: 48 8d 05 92 2e 00 00 lea 0x2e92(%rip),%rax    # 4010 <__TMC_END__>
117e: 48 39 f8                cmp  %rdi,%rax
1181: 74 15                  je   1198 <deregister_tm_clones+0x28>
1183: 48 8b 05 4e 2e 00 00 mov  0x2e4e(%rip),%rax    # 3fd8
```

<_ITM_deregisterTMCloneTable>

```
118a: 48 85 c0                test %rax,%rax
118d: 74 09                  je   1198 <deregister_tm_clones+0x28>
118f: ff e0                  jmpq *%rax
1191: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)
1198: c3                      retq
1199: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)
```

00000000000011a0 <register_tm_clones>:

```
11a0: 48 8d 3d 69 2e 00 00 lea 0x2e69(%rip),%rdi    # 4010 <__TMC_END__>
11a7: 48 8d 35 62 2e 00 00 lea 0x2e62(%rip),%rsi    # 4010 <__TMC_END__>
11ae: 48 29 fe                sub  %rdi,%rsi
11b1: 48 89 f0                mov  %rsi,%rax
11b4: 48 c1 ee 3f            shr  $0x3f,%rsi
11b8: 48 c1 f8 03            sar  $0x3,%rax
11bc: 48 01 c6                add  %rax,%rsi
11bf: 48 d1 fe                sar  %rsi
11c2: 74 14                  je   11d8 <register_tm_clones+0x38>
11c4: 48 8b 05 25 2e 00 00 mov  0x2e25(%rip),%rax    # 3ff0
```

<_ITM_registerTMCloneTable>

```
11cb: 48 85 c0                test %rax,%rax
11ce: 74 08                  je   11d8 <register_tm_clones+0x38>
11d0: ff e0                  jmpq *%rax
11d2: 66 0f 1f 44 00 00 00 nopw 0x0(%rax,%rax,1)
11d8: c3                      retq
11d9: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)
```

00000000000011e0 <__do_global_dtors_aux>:

```
11e0: f3 0f 1e fa            endbr64
11e4: 80 3d 25 2e 00 00 00 cmpb $0x0,0x2e25(%rip)    # 4010 <__TMC_END__>
11eb: 75 2b                  jne  1218 <__do_global_dtors_aux+0x38>
11ed: 55                      push %rbp
11ee: 48 83 3d 02 2e 00 00 cmpq $0x0,0x2e02(%rip)    # 3ff8
```

<__cxa_finalize@GLIBC_2.2.5>

```
11f5: 00
11f6: 48 89 e5                mov  %rsp,%rbp
11f9: 74 0c                  je   1207 <__do_global_dtors_aux+0x27>
11fb: 48 8b 3d 06 2e 00 00 mov  0x2e06(%rip),%rdi    # 4008 <__dso_handle>
1202: e8 a9 fe ff ff          callq 10b0 <__cxa_finalize@plt>
1207: e8 64 ff ff ff          callq 1170 <deregister_tm_clones>
120c: c6 05 fd 2d 00 00 01 movb $0x1,0x2dfd(%rip)    # 4010 <__TMC_END__>
1213: 5d                      pop  %rbp
1214: c3                      retq
```

```

1215:    0f 1f 00          nopl  (%rax)
1218:    c3                retq
1219:    0f 1f 80 00 00 00 00 nopl  0x0(%rax)

```

0000000000001220 <frame_dummy>:

```

1220:    f3 0f 1e fa      endbr64
1224:    e9 77 ff ff      jmpq  11a0 <register_tm_clones>

```

0000000000001229 <main>:

//hello.c

#include <stdio.h>

#include <stdlib.h>

int main (int argc, char **argv)

{

```

1229:    f3 0f 1e fa      endbr64
122d:    55                push  %rbp
122e:    48 89 e5          mov   %rsp,%rbp
1231:    48 83 c4 80      add   $0xfffffffffff80,%rsp
1235:    89 7d 8c          mov   %edi,-0x74(%rbp)
1238:    48 89 75 80      mov   %rsi,-0x80(%rbp)
123c:    64 48 8b 04 25 28 00 mov   %fs:0x28,%rax
1243:    00 00
1245:    48 89 45 f8      mov   %rax,-0x8(%rbp)
1249:    31 c0            xor   %eax,%eax

```

char c1;

unsigned char c2;

int i1=0;

```

124b:    c7 45 9c 00 00 00 00 movl  $0x0,-0x64(%rbp)

```

long i2=0;

```

1252:    48 c7 45 a0 00 00 00 00 movq  $0x0,-0x60(%rbp)
1259:    00

```

char *cptr;

int *iptr;

long *lptr;

char array1[40] = "This is a string";

```

125a:    48 b8 54 68 69 73 20 movabs $0x2073692073696854,%rax
1261:    69 73 20
1264:    48 ba 61 20 73 74 72 movabs $0x676e697274732061,%rdx
126b:    69 6e 67
126e:    48 89 45 d0      mov   %rax,-0x30(%rbp)
1272:    48 89 55 d8      mov   %rdx,-0x28(%rbp)
1276:    48 c7 45 e0 00 00 00 00 movq  $0x0,-0x20(%rbp)
127d:    00
127e:    48 c7 45 e8 00 00 00 00 movq  $0x0,-0x18(%rbp)
1285:    00
1286:    48 c7 45 f0 00 00 00 00 movq  $0x0,-0x10(%rbp)
128d:    00

```

```

cptr = (char *)malloc(200);
128e:    bf c8 00 00 00    mov     $0xc8,%edi
1293:    e8 88 fe ff ff    callq   1120 <malloc@plt>
1298:    48 89 45 a8      mov     %rax,-0x58(%rbp)
iptr = (int *)malloc(200);
129c:    bf c8 00 00 00    mov     $0xc8,%edi
12a1:    e8 7a fe ff ff    callq   1120 <malloc@plt>
12a6:    48 89 45 b0      mov     %rax,-0x50(%rbp)
lptr = (long *)malloc(200);
12aa:    bf c8 00 00 00    mov     $0xc8,%edi
12af:    e8 6c fe ff ff    callq   1120 <malloc@plt>
12b4:    48 89 45 b8      mov     %rax,-0x48(%rbp)

c1 = 'X';
12b8:    c6 45 99 58      movb    $0x58,-0x67(%rbp)
c2 = 0x44;
12bc:    c6 45 9a 44      movb    $0x44,-0x66(%rbp)
i1 = 0x100;
12c0:    c7 45 9c 00 01 00 00 movl    $0x100,-0x64(%rbp)
l2 = 0x0123456789abcdef;
12c7:    48 b8 ef cd ab 89 67 movabs  $0x123456789abcdef,%rax
12ce:    45 23 01
12d1:    48 89 45 a0      mov     %rax,-0x60(%rbp)

*iptr = 0x2000;
12d5:    48 8b 45 b0      mov     -0x50(%rbp),%rax
12d9:    c7 00 00 20 00 00 movl    $0x2000,(%rax)
*lptr = 0x88889999aaaabbbb;
12df:    48 8b 45 b8      mov     -0x48(%rbp),%rax
12e3:    48 b9 bb bb aa aa 99 movabs  $0x88889999aaaabbbb,%rcx
12ea:    99 88 88
12ed:    48 89 08      mov     %rcx,(%rax)

printf("Hello World\n");
12f0:    48 8d 3d 0d 0d 00 00 lea     0xd0d(%rip),%rdi    # 2004 <_IO_stdin_used+0x4>
12f7:    e8 c4 fd ff ff    callq   10c0 <puts@plt>
printf("\n\n");
12fc:    48 8d 3d 0d 0d 00 00 lea     0xd0d(%rip),%rdi    # 2010 <_IO_stdin_used+0x10>
1303:    e8 b8 fd ff ff    callq   10c0 <puts@plt>
printf("l2: %lx \n", l2);
1308:    48 8b 45 a0      mov     -0x60(%rbp),%rax
130c:    48 89 c6      mov     %rax,%rsi
130f:    48 8d 3d fc 0c 00 00 lea     0xcfc(%rip),%rdi    # 2012 <_IO_stdin_used+0x12>
1316:    b8 00 00 00 00    mov     $0x0,%eax
131b:    e8 d0 fd ff ff    callq   10f0 <printf@plt>
printf("i1: %x \n", i1);
1320:    8b 45 9c      mov     -0x64(%rbp),%eax
1323:    89 c6      mov     %eax,%esi
1325:    48 8d 3d f0 0c 00 00 lea     0xcf0(%rip),%rdi    # 201c <_IO_stdin_used+0x1c>

```

```

132c:    b8 00 00 00 00    mov    $0x0,%eax
1331:    e8 ba fd ff ff    callq 10f0 <printf@plt>
printf("i1: %10x \n", i1);
1336:    8b 45 9c          mov    -0x64(%rbp),%eax
1339:    89 c6             mov    %eax,%esi
133b:    48 8d 3d e3 0c 00 00 lea    0xce3(%rip),%rdi    # 2025 <_IO_stdin_used+0x25>
1342:    b8 00 00 00 00    mov    $0x0,%eax
1347:    e8 a4 fd ff ff    callq 10f0 <printf@plt>
printf("i1: %4x \n", i1);
134c:    8b 45 9c          mov    -0x64(%rbp),%eax
134f:    89 c6             mov    %eax,%esi
1351:    48 8d 3d d8 0c 00 00 lea    0xcd8(%rip),%rdi    # 2030 <_IO_stdin_used+0x30>
1358:    b8 00 00 00 00    mov    $0x0,%eax
135d:    e8 8e fd ff ff    callq 10f0 <printf@plt>
printf("c1: %c \n", c1);
1362:    0f be 45 99       movsbl -0x67(%rbp),%eax
1366:    89 c6             mov    %eax,%esi
1368:    48 8d 3d cb 0c 00 00 lea    0xccb(%rip),%rdi    # 203a <_IO_stdin_used+0x3a>
136f:    b8 00 00 00 00    mov    $0x0,%eax
1374:    e8 77 fd ff ff    callq 10f0 <printf@plt>
printf("string: %s \n", array1);
1379:    48 8d 45 d0       lea    -0x30(%rbp),%rax
137d:    48 89 c6          mov    %rax,%rsi
1380:    48 8d 3d bc 0c 00 00 lea    0xcbc(%rip),%rdi    # 2043 <_IO_stdin_used+0x43>
1387:    b8 00 00 00 00    mov    $0x0,%eax
138c:    e8 5f fd ff ff    callq 10f0 <printf@plt>

```

/*Copy from input.txt to output.txt */

```

FILE *input = fopen("input.txt","r");
1391:    48 8d 35 b8 0c 00 00 lea    0xcb8(%rip),%rsi    # 2050 <_IO_stdin_used+0x50>
1398:    48 8d 3d b3 0c 00 00 lea    0xcb3(%rip),%rdi    # 2052 <_IO_stdin_used+0x52>
139f:    e8 8c fd ff ff    callq 1130 <fopen@plt>
13a4:    48 89 45 c0       mov    %rax,-0x40(%rbp)
FILE *output = fopen("output.txt", "w");
13a8:    48 8d 35 ad 0c 00 00 lea    0xcad(%rip),%rsi    # 205c <_IO_stdin_used+0x5c>
13af:    48 8d 3d a8 0c 00 00 lea    0xca8(%rip),%rdi    # 205e <_IO_stdin_used+0x5e>
13b6:    e8 75 fd ff ff    callq 1130 <fopen@plt>
13bb:    48 89 45 c8       mov    %rax,-0x38(%rbp)

```

char ch;

while((ch = fgetc(input)) != EOF)

```

13bf:    eb 12             jmp    13d3 <main+0x1aa>
    fputc(ch, output);
13c1:    0f be 45 9b       movsbl -0x65(%rbp),%eax
13c5:    48 8b 55 c8       mov    -0x38(%rbp),%rdx
13c9:    48 89 d6          mov    %rdx,%rsi
13cc:    89 c7             mov    %eax,%edi
13ce:    e8 3d fd ff ff    callq 1110 <fputc@plt>

```

while((ch = fgetc(input)) != EOF)

```
13d3: 48 8b 45 c0      mov  -0x40(%rbp),%rax
13d7: 48 89 c7         mov  %rax,%rdi
13da: e8 21 fd ff ff   callq 1100 <fgetc@plt>
13df: 88 45 9b         mov  %al,-0x65(%rbp)
13e2: 80 7d 9b ff     cmpb $0xff,-0x65(%rbp)
13e6: 75 d9          jne  13c1 <main+0x198>
```

fclose(input);

```
13e8: 48 8b 45 c0      mov  -0x40(%rbp),%rax
13ec: 48 89 c7         mov  %rax,%rdi
13ef: e8 dc fc ff ff   callq 10d0 <fclose@plt>
```

fclose(output);

```
13f4: 48 8b 45 c8      mov  -0x38(%rbp),%rax
13f8: 48 89 c7         mov  %rax,%rdi
13fb: e8 d0 fc ff ff   callq 10d0 <fclose@plt>
```

return 0;

```
1400: b8 00 00 00 00   mov  $0x0,%eax
}
1405: 48 8b 4d f8      mov  -0x8(%rbp),%rcx
1409: 64 48 33 0c 25 28 00 xor  %fs:0x28,%rcx
1410: 00 00
1412: 74 05          je   1419 <main+0x1f0>
1414: e8 c7 fc ff ff   callq 10e0 <__stack_chk_fail@plt>
1419: c9             leaveq
141a: c3             retq
141b: 0f 1f 44 00 00   nopl 0x0(%rax,%rax,1)
```

0000000000001420 <__libc_csu_init>:

```
1420: f3 0f 1e fa     endbr64
1424: 41 57           push  %r15
1426: 4c 8d 3d 53 29 00 00 lea  0x2953(%rip),%r15    # 3d80
<__frame_dummy_init_array_entry>
142d: 41 56           push  %r14
142f: 49 89 d6        mov  %rdx,%r14
1432: 41 55           push  %r13
1434: 49 89 f5        mov  %rsi,%r13
1437: 41 54           push  %r12
1439: 41 89 fc        mov  %edi,%r12d
143c: 55             push  %rbp
143d: 48 8d 2d 44 29 00 00 lea  0x2944(%rip),%rbp    # 3d88
<__do_global_ctors_aux_fini_array_entry>
1444: 53             push  %rbx
1445: 4c 29 fd        sub  %r15,%rbp
1448: 48 83 ec 08     sub  $0x8,%rsp
144c: e8 af fb ff ff   callq 1000 <_init>
1451: 48 c1 fd 03     sar  $0x3,%rbp
```

```

1455:    74 1f                je    1476 <__libc_csu_init+0x56>
1457:    31 db                xor    %ebx,%ebx
1459:    0f 1f 80 00 00 00 00 nopl  0x0(%rax)
1460:    4c 89 f2             mov    %r14,%rdx
1463:    4c 89 ee             mov    %r13,%rsi
1466:    44 89 e7             mov    %r12d,%edi
1469:    41 ff 14 df          callq *(%r15,%rbx,8)
146d:    48 83 c3 01          add    $0x1,%rbx
1471:    48 39 dd             cmp    %rbx,%rbp
1474:    75 ea                jne    1460 <__libc_csu_init+0x40>
1476:    48 83 c4 08          add    $0x8,%rsp
147a:    5b                   pop    %rbx
147b:    5d                   pop    %rbp
147c:    41 5c                pop    %r12
147e:    41 5d                pop    %r13
1480:    41 5e                pop    %r14
1482:    41 5f                pop    %r15
1484:    c3                   retq
1485:    66 66 2e 0f 1f 84 00 data16 nopw %cs:0x0(%rax,%rax,1)
148c:    00 00 00 00

```

0000000000001490 <__libc_csu_fini>:

```

1490:    f3 0f 1e fa          endbr64
1494:    c3                   retq

```

Disassembly of section .fini:

0000000000001498 <_fini>:

```

1498:    f3 0f 1e fa          endbr64
149c:    48 83 ec 08          sub    $0x8,%rsp
14a0:    48 83 c4 08          add    $0x8,%rsp
14a4:    c3                   retq

```