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

SYMBOL TABLE:
0000000000000318 l    d  .interp        0000000000000000              .interp
0000000000000338 l    d  .note.gnu.property         0000000000000000              .note.gnu.property
0000000000000358 l    d  .note.gnu.build-id0000000000000000              .note.gnu.build-id
000000000000037c l    d  .note.ABI-tag        0000000000000000              .note.ABI-tag
00000000000003a0 l    d  .gnu.hash  0000000000000000              .gnu.hash
00000000000003c8 l    d  .dynsym    0000000000000000              .dynsym
0000000000000518 l    d  .dynstr    0000000000000000              .dynstr
00000000000005dc l    d  .gnu.version         0000000000000000              .gnu.version
00000000000005f8 l    d  .gnu.version_r       0000000000000000              .gnu.version_r
0000000000000628 l    d  .rela.dyn  0000000000000000              .rela.dyn
00000000000006e8 l    d  .rela.plt  0000000000000000              .rela.plt
0000000000001000 l    d  .init      0000000000000000              .init
0000000000001020 l    d  .plt       0000000000000000              .plt
00000000000010b0 l    d  .plt.got   0000000000000000              .plt.got
00000000000010c0 l    d  .plt.sec   0000000000000000              .plt.sec
0000000000001140 l    d  .text      0000000000000000              .text
0000000000001498 l    d  .fini      0000000000000000              .fini
0000000000002000 l    d  .rodata    0000000000000000              .rodata
000000000000206c l    d  .eh_frame_hdr        0000000000000000              .eh_frame_hdr
00000000000020b0 l    d  .eh_frame 0000000000000000              .eh_frame
0000000000003d80 l    d  .init_array0000000000000000              .init_array
0000000000003d88 l    d  .fini_array0000000000000000              .fini_array
0000000000003d90 l    d  .dynamic  0000000000000000              .dynamic
0000000000003f80 l    d  .got       0000000000000000              .got
0000000000004000 l    d  .data      0000000000000000              .data
```

```
0000000000004010 l    d  .bss          0000000000000000           .bss
0000000000000000 l    d  .comment 0000000000000000           .comment
0000000000000000 l    d  .debug_aranges    0000000000000000           .debug_aranges
0000000000000000 l    d  .debug_info       0000000000000000           .debug_info
0000000000000000 l    d  .debug_abbrev     0000000000000000           .debug_abbrev
0000000000000000 l    d  .debug_line       0000000000000000           .debug_line
0000000000000000 l    d  .debug_str0000000000000000           .debug_str
0000000000000000 l    df *ABS*     0000000000000000           crtstuff.c
0000000000001170 l     F .text         0000000000000000           deregister_tm_clones
00000000000011a0 l     F .text         0000000000000000           register_tm_clones
00000000000011e0 l     F .text         0000000000000000           __do_global_dtors_aux
0000000000004010 l     O .bss          0000000000000001           completed.8060
0000000000003d88 l     O .fini_array         0000000000000000
__do_global_dtors_aux_fini_array_entry
0000000000001220 l     F .text         0000000000000000           frame_dummy
0000000000003d80 l     O .init_array         0000000000000000
__frame_dummy_init_array_entry
0000000000000000 l    df *ABS*     0000000000000000           hello.c
0000000000000000 l    df *ABS*     0000000000000000           crtstuff.c
00000000000021b4 l     O .eh_frame          0000000000000000           __FRAME_END__
0000000000000000 l    df *ABS*     0000000000000000
0000000000003d88 l       .init_array 0000000000000000           __init_array_end
0000000000003d90 l     O .dynamic 0000000000000000           _DYNAMIC
0000000000003d80 l       .init_array 0000000000000000           __init_array_start
000000000000206c l       .eh_frame_hdr      0000000000000000           __GNU_EH_FRAME_HDR
0000000000003f80 l     O .got          0000000000000000           _GLOBAL_OFFSET_TABLE_
0000000000001000 l     F .init         0000000000000000           _init
0000000000001490 g     F .text         0000000000000005           __libc_csu_fini
0000000000000000  w      *UND*    0000000000000000           _ITM_deregisterTMCloneTable
0000000000004000  w      .data         0000000000000000           data_start
0000000000000000       F *UND*    0000000000000000           puts@@GLIBC_2.2.5
0000000000004010 g       .data         0000000000000000           _edata
0000000000000000       F *UND*    0000000000000000           fclose@@GLIBC_2.2.5
0000000000001498 g     F .fini         0000000000000000           .hidden _fini
0000000000000000       F *UND*    0000000000000000           __stack_chk_fail@@GLIBC_2.4
0000000000000000       F *UND*    0000000000000000           printf@@GLIBC_2.2.5
0000000000000000       F *UND*    0000000000000000           fgetc@@GLIBC_2.2.5
0000000000000000       F *UND*    0000000000000000           fputc@@GLIBC_2.2.5
0000000000000000       F *UND*    0000000000000000           __libc_start_main@@GLIBC_2.2.5
0000000000004000 g       .data         0000000000000000           __data_start
0000000000000000  w      *UND*    0000000000000000           __gmon_start__
0000000000004008 g     O .data         0000000000000000           .hidden __dso_handle
0000000000002000 g     O .rodata       0000000000000004           _IO_stdin_used
0000000000001420 g     F .text         0000000000000065           __libc_csu_init
0000000000000000       F *UND*    0000000000000000           malloc@@GLIBC_2.2.5
0000000000004018 g       .bss          0000000000000000           _end
0000000000001140 g     F .text         000000000000002f           _start
0000000000004010 g       .bss          0000000000000000           __bss_start
0000000000001229 g     F .text         00000000000001f2           main
```

```
0000000000000000       F *UND*    0000000000000000              fopen@@GLIBC_2.2.5
0000000000004010 g     O .data    0000000000000000              .hidden __TMC_END__
0000000000000000  w      *UND*    0000000000000000              _ITM_registerTMCloneTable
0000000000000000  w    F *UND*    0000000000000000              __cxa_finalize@@GLIBC_2.2.5
```