ICS Homework 3

March 21, 2020

1 System Software

1.1 Fork and Execve

Read the C program and answer the questions below. NOTE: /bin/echo is an executable file that will print its arguments on the screen.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

char *ch;

int main(void)
{
  ch = malloc(1);
  *ch = 'A';

  if (fork() == 0) {
    *ch = 'B';
    printf("%c\n", *ch);

    if (fork() == 0)
      printf("C\n");
    else
      exit(0);
  } else {
    while (waitpid(-1, NULL,
                   WUNTRACED) > 0);
    char *my_argv[] = {"/bin/echo", ch, 0};
    execve(my_argv[0], my_argv, 0);
    printf("D\n");
  }
  free(ch);
  return 0;
}
```

1. What is the possible output of this program? Is the output deterministic? Please explain why.

   The output of this program is not deterministic. It can be `B\nC\nA\n` or `B\nA\nC\n`. `D` will not appear due to that `execve` will never return.

2. What if we print the address of `ch` in all processes? Will them the be same? Will their value be the same? Why?

```
/* same as before */
  ch = malloc(1);
  *ch = 'A';

  if (fork() == 0) {
    *ch = 'B';
    printf("%p:%c\n", ch, *ch);
    if (fork() == 0)
        *ch = 'C';
        printf("%p:%c\n", ch, *ch);
    else
        exit(0);
  } else {
        printf("%p:%c\n", ch, *ch);
    while (waitpid(-1, NULL,
        WUNTRACED) > 0);
/* same as before */
```

   The address of `ch` will be the same, but the value won't. It is because that even though they share the same virtual address, they are mapped to different physical memory.

3. Is there any memory leakage or double free issue for the variable ch in each process? Please explain why.

   There is no double free issue since the memory spaces of these three processes are isolated.

   The grandchild process frees the allocated memory before exits while the child process does not. The parent process calls execve and afterwards its memory mappings are not preserved (Modern implementations of malloc() use anonymous memory mappings, see the man page of mmap(2) and execve(2)) so no memory leakage for him.

2 Organization

2.1 Pipeline

1. Please write down the HCL code for the following signals in PIPE implementation. NOTE: You should refer to Chapter 4.5 of CSAPP book.

a. d_valB

ANS:

```
word d_valB = [
  d_srcB == e_dstE : e_valE;
  d_srcB == M_dstM : m_valM;
  d_srcB == M_dstE : M_valE;
  d_srcB == W_dstM : W_valM;
  d_srcB == W_dstE: W_ValE;
  1 : d_rvalB;
]
```

b. D_stall

ANS:

```
bool D_stall =
  E_icode in { IMRMOVQ, IPOPQ } &&
  E_dstM in { dsrcA, dsrcB };
```

c. E_bubble

ANS:

```
bool E_bubble =
  (E_icode == IJXX && !e_Cnd) ||
  (E_icode in { IMRMOVQ, IPOPQ } &&
   E_dstM in { d_srcA, d_srcB });
```

2. What's the difference between signal e_dstE and E_dstE? When are they updated?

   ANS: e_dstE is the signal generated and updated in Execute stage, while E_dstE is in the Pipeline Register updated during a rising clock.