PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

1ra práctica (tipo a)
(Primer semestre de 2015)

Horario 0781: prof. V. Khlebnikov

Duración: 1 h. 50 min.
Nota: No se puede usar ningún material de consulta.
La presentación, la ortografía y la gramática influirán en la calificación.
Puntaje total: 20 puntos

Pregunta 1 (4 puntos – 20 min.) (*OSIDP7E, Chapter 1*) Suppose the hypothetical processor of Figure 1.3 also has two I/O instructions:

0011 = Load AC from I/O
0111 = Store AC to I/O

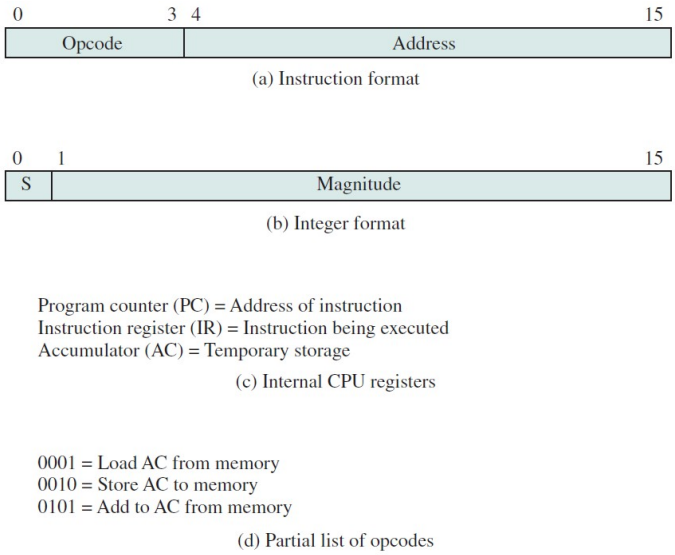

Figure 1.3 Characteristics of a Hypothetical Machine

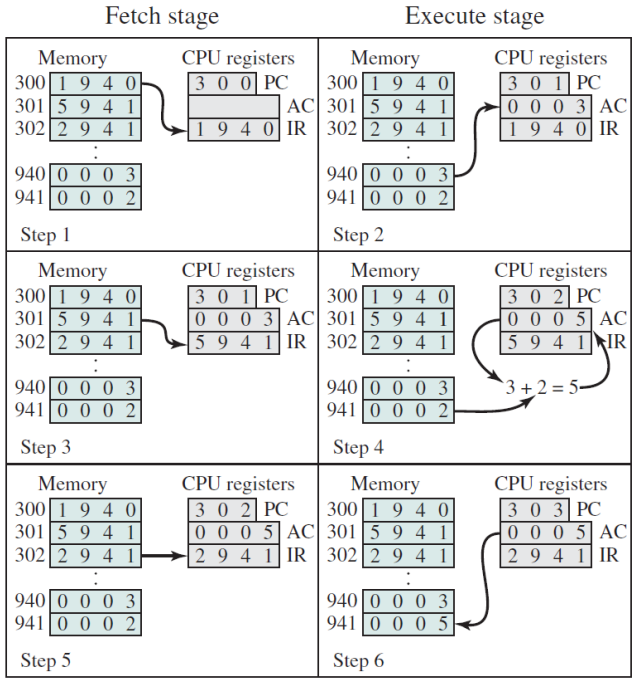

Figure 1.4 Example of Program Execution (contents of memory and registers in hexadecimal)

In these cases, the 12-bit address identifies a particular external device. Show the program execution (using format of Figure 1.4) for following program:

1. Load AC from device 5.
2. Add contents of memory location 940.
3. Store AC to device 6.

Assume that the next value retrieved from device 5 is 3 and that location 940 contains a value of 2.

Pregunta 2 (1 punto – 5 min.) A computer has 1 GB of RAM of which the operating system occupies 512 MB. The processes are all 256 MB (for simplicity) and have the same characteristics. If the goal is 99% CPU utilization, what is the maximum I/O wait that can be tolerated?

Pregunta 3 **(3 puntos – 15 min.)** ¿Cuál es la diferencia entre las salidas estándar de dos programas con los siguientes códigos?

```
void
func_A()
{
    pid_t pid[N];
    int i, status;

    for (i = 0; i < N; i++)
        if ((pid[i] = fork()) == 0) {
            exit(100+i);
        }
    for (i = 0; i < N; i++) {
        pid_t wpid = wait(&status);
        if (WIFEXITED(status))
            printf("%d terminated with exit status %d\n", wpid, WEXITSTATUS(status));
        else
            printf("%d terminate abnormally\n", wpid);
    }
}

void
func_B()
{
    pid_t pid[N];
    int i;
    int status;

    for (i = 0; i < N; i++)
        if ((pid[i] = fork()) == 0)
            exit(100+i);
    for (i = 0; i < N; i++) {
        pid_t wpid = waitpid(pid[i], &status, 0);
        if (WIFEXITED(status))
            printf("%d terminated with exit status %d\n", wpid, WEXITSTATUS(status));
        else
            printf("%d terminate abnormally\n", wpid);
    }
}
```

Pregunta 4 **(4 puntos – 20 min.)** ¿Cuál será la salida del siguiente programa?

```
$ cat 2015-1_pr1.c

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int
main(void)
{
  char weeks[]="0279";
  char *w=weeks;
  char p[]="/bin/date";
  char d[]="--date=. weeks Fri 11";
  int i=1;

  while (d[7]=*w) {
    fork()? waitpid(-1,NULL,0) : execl(p,&p[5],d,(char *)0);
    w++;
  }
  exit(0);
}

$ date
vie abr 10 11:10:00 PET 2015

$ date "--date=265 days"
jue dic 31 11:10:00 PET 2015

$ date "--date=100 days ago"
mié dic 31 11:10:00 PET 2014

$ date "--date=265 days 23:59"
jue dic 31 23:59:00 PET 2015

$ ./2015-1_pr1
...
```

Pregunta 5 (4 puntos – 20 min.) (Minix3) La función `test7a()` presentada invoca a la función de procesamiento de errores `e()`. Explique cuáles son las 7 situaciones que se consideran erróneas (**3,5 puntos**). Explique por qué el valor inicial de la variable `i` (la línea `23`) es `3` (**0,5 puntos**).

```
$ cat -n test7a..c
```

```
            #define ITEMS  32

            char buf[ITEMS] = {0,1,2,3,4,5,6,7,8,9,8,7,6,5,4,3,2,1,0,1,2,3,4,5,6,7,8,9};

     1      void test7a()
     2      {
     3        int i, fd[2], ect;
     4        char buf2[ITEMS+1];
     5
     6        subtest = 1;
     7        if (pipe(fd) != 0) e(1);
     8        if (write(fd[1], buf, ITEMS) != ITEMS) e(2);
     9        buf2[0] = 0;
    10        if (read(fd[0], buf2, ITEMS+1) != ITEMS) e(3);
    11        ect = 0;
    12        for (i = 0; i < ITEMS; i++) if (buf[i] != buf2[i]) ect++;
    13        if (ect != 0) e(4);
    14        if (close(fd[0]) != 0) e(5);
    15        if (close(fd[1]) != 0) e(6);
    16
    17        errno = 0;
    18        while (1) {
    19              if (pipe(fd) < 0) break;
    20        }
    21        if (errno != EMFILE) e(7);
    22
    23        for (i = 3; i < OPEN_MAX; i++) close(i);
    24      }
```

Pregunta 6 (4 puntos – 20 min.) (Minix3) Justifique el siguiente código explicando el objetivo de cada bloque.

```
$ cat -n test2a..c
```

```
            char buf[2048];

     1      void test2a()
     2      {
     3        int fd[2];
     4        int n, i, j, q = 0;
     5
     6        subtest = 1;
     7        if (pipe(fd) < 0) {
     8              errct++;
     9              quit();
    10        }
    11        i = fork();
    12        if (i < 0) {
    13              errct++;
    14              quit();
    15        }
    16        if (i != 0) {
    17              close(fd[0]);
    18              for (i = 0; i < 2048; i++) buf[i] = i & 0377;
    19              for (q = 0; q < 8; q++) {
    20                      if (write(fd[1], buf, 2048) < 0) {
    21                              errct++;
    22                              quit();
    23                      }
    24              }
    25              close(fd[1]);
    26              wait(&q);
    27              if (q != 256 * 58) {
    28                      errct++;
    29                      quit();
    30              }
```

```
31          } else {
32                close(fd[1]);
33                for (q = 0; q < 32; q++) {
34                        n = read(fd[0], buf, 512);
35                        if (n != 512) {
36                                errct++;
37                                quit();
38                        }
39                        for (j = 0; j < n; j++)
40                                if ((buf[j] & 0377) != (kk & 0377)) {
41                                        printf("... %d %d %d \n ",
42                                                        j, buf[j] & 0377, kk & 0377);
43                                } else {
44                                        kk++;
45                                }
46                }
47                exit(58);
48          }
49        }
```

La práctica ha sido preparada por VK
con LibreOffice Writer en Linux Mint 17.1.

Profesor del curso: (0781) V. Khlebnikov

Pando, 10 de abril de 2015