

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

Ira práctica (tipo a)
(Segundo semestre de 2020)

Horario 0781: prof. V. Khlebnikov

Horario 0782: prof. F. Solari A.

Duración: 1 h. 50 min.

Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**

Puntaje total: 20 puntos

Pregunta 1 (6 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 1** de la **Práctica 1** en PAIDEIA **antes de las 11:40**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_12.txt`. Por ejemplo, `20171903_12.txt`.

Si en un programa está la siguiente sentencia:

```
while ((fork()*fork()) && --n) { ... };
```

donde `int n = 3;`

(a) (2 puntos) ¿Cuántos procesos y cuántas veces ejecutarán el cuerpo del bucle?

(b) (4 puntos) ¿Cuál será el árbol de procesos creados? Solamente indique la cantidad de nodos del árbol en cada nivel y la cantidad de hijos que tiene cada padre.

Pregunta 2 (6 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 2** de la **Práctica 1** en PAIDEIA **antes de las 12:20**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser `<su_código_de_8_dígitos>_12.txt`. Por ejemplo, `20171903_12.txt`.

Dado el siguiente segmento de código, mostrado mediante `cat -n`:

```
$ cat -n preg_p1_2.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <wait.h>
5
6  int main(int argc, char *argv[]) {
7      int filedes[2];
8      char outfile[64];
9      int p,status;
10
11      if(pipe(filedes)==-1) { perror("Error en pipe");exit(-1);}
12      if(fork()==0) { /* estamos en el hijo */
13          usleep(6000);
14          close(filedes[1]); /* no va a escribir */
15          dup2(filedes[0],0); /* 0 es la STDINPUT_FILENO */
16          close(filedes[0]); /* ya esta duplicado el filedes[0] */
17          sprintf(outfile,"%s.out",argv[0]);
18          execlp("tee","tee",outfile,NULL);
19          perror("Error en exec -2");
20          exit(-2);
21      }
22      else {
23          if(fork()==0) {
24              system("ps lf");
25              system("echo linea separadora;echo");
26              close(filedes[0]); /* no va a leer */
27              dup2(filedes[1],1); /* 0 es la STDOUT_FILENO */
28              close(filedes[1]); /* ya esta duplicado el filedes[1] */
29              execlp("ps","ps", "lf", NULL);
30              perror("Error en exec -3");
31              exit(-3);
32          }
33      }
34      close(filedes[0]);close(filedes[1]);
35      while ((p=waitpid(-1,&status,0))>0)
36          fprintf(stderr,"waitpid %d, status %d\n",p,WEXITSTATUS(status));
37  }
```

```

38     exit(0);
39 }
40

```

La ejecución del programa muestra en la terminal lo siguiente:

```

$ ./preg_p1_2 | cat -n
 1 F   UID      PID      PPID PRI  NI       VSZ      RSS WCHAN  STAT TTY      TIME COMMAND
 2 0   1001     1418     1412  20   0    13360   5224 do_wai  Ss   pts/0    0:00 bash
 3 0   1001     9723     1418  20   0    2356    516 do_wai  S+   pts/0    0:00 \_  ./preg_p1_2
 4 1   1001     9725     9723  20   0    2356     76 hrtime  S+   pts/0    0:00 |   \_  ./preg_p1_2
 5 1   1001     9726     9723  20   0    2356    532 do_wai  S+   pts/0    0:00 |   \_  ./preg_p1_2
 6 0   1001     9727     9726  20   0    2608    600 do_wai  S+   pts/0    0:00 |       \_  sh -c ps lf
 7 0   1001     9728     9727  20   0   13936   3216 -       R+   pts/0    0:00 |       \_  ps lf
 8 0   1001     9724     1418  20   0   11400    580 pipe_w  S+   pts/0    0:00 \_  cat -n
 9 línea separadora
10
11 F   UID      PID      PPID PRI  NI       VSZ      RSS WCHAN  STAT TTY      TIME COMMAND
12 0   1001     1418     1412  20   0    13360   5224 do_wai  Ss   pts/0    0:00 bash
13 0   1001     9723     1418  20   0    2356    516 do_wai  S+   pts/0    0:00 \_  ./preg_p1_2
14 0   1001     9725     9723  20   0   10620    520 pipe_w  S+   pts/0    0:00 |   \_  tee ./preg_p1_2.out
15 0   1001     9726     9723  20   0   13936   3248 -       R+   pts/0    0:00 |       \_  ps lf
16 0   1001     9724     1418  20   0   11400    580 pipe_w  S+   pts/0    0:00 \_  cat -n
waitpid 9725, status 0
waitpid 9726, status 0

```

a) (2 puntos) Haga un esquema simple (con caracteres como los mostrados para los procesos, y “->” o “<-” como flechas de escritura y lectura para las tuberías), que muestre los procesos y su relación padre-hijo, así como las conexiones finales entre ellos al momento de hacer *execlp()* y *waitpid()*. Detalle con los números de proceso, números de los descriptores, y nombres de programa.

b) (1 punto) ¿Por qué las líneas “waitpid xxx, status y” no están numeradas?

c) (3 puntos) Indique cómo puede modificarse el programa para que *cat -n* numere solo la salida que va al resultado en el archivo “preg_p1_p2.out”, haciendo primero un esquema de los procesos y programas involucrados, así como de sus conexiones y estableciendo que proceso sería el responsable de los recursos adicionales, y alguna otra modificación. No tiene que escribir el nuevo programa.

Pregunta 3 (8 puntos – 30 min.) Su respuesta debe estar en la carpeta **Buzón 3** de la **Práctica 1** en PAIDEIA antes de las **13:00**. Por cada 3 minutos de retardo son -2 puntos.

El nombre de su archivo debe ser <su_código_de_8_dígitos>_13.txt. Por ejemplo, 20171903_13.txt.

a) (4 puntos) En los conceptos de Sistemas Operativos, responda brevemente las siguientes cuestiones, justificando su respuesta en cada caso:

1. (1 punto) La llamada al sistema *pipe()*, corresponde a ¿cuál concepto de sistema operativo?
2. (1 punto) ¿Por qué es necesario que el hardware de CPU provea alguna instrucción como *trap*, *swi*, *int*, *system*?
3. (1 punto) Respecto a la estructura de *microkernel*, señale una ventaja y una desventaja de su implementación.
4. (1 punto) Los entornos de *virtualización* como VirtualBox o VMware Workstation/Player, ¿cómo se clasifican en cuanto a tipo de *Hypervisor*, *tipo I* o *tipo II*? ¿Qué ocurre cuando un proceso en un invitado hace una llamada al sistema que requiere acceso al hardware? (Describa brevemente los diferentes niveles que debe recorrerse para completarla)

b) (4 puntos) Considere dos procesos, *A* y *B*, que se ejecutan concurrentemente compartiendo la variable *x*.

	Proceso A	Proceso B
A1	<i>x</i> := 3 <i>x</i> ;	B1: <i>x</i> := 10;
A2	<i>x</i> := 5 <i>x</i> ;	B2: <i>x</i> := <i>x</i> + 2;

Suponiendo que la ejecución de cada línea de código es atómica, ¿cuáles son los posibles valores finales de la variable *x* si su valor inicial es 0?



La práctica ha sido preparada por FS (2,3a) y VK (1,3b) con LibreOffice Writer en Linux Mint 20 “Ulyana”

Profesores del curso: (0781) V. Khlebnikov
(0782) F. Solari A.

Lima, 2 de octubre de 2020