

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

1ra práctica (tipo a)
(Primer semestre de 2014)

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) (*AST – MOS3E, Chapter 1*)

a) (1 punto – 5 min.) There is a wide variation in the nomenclature. On some computers this term refers to any interrupt, on some machines to any synchronous interrupt, on some machines to any interrupt not associated with input/output, on some machines only to interrupts caused by some special instructions, etc.

(0,5 puntos) What is this term?

(0,5 puntos) What is the purpose of this special instruction?

b) (3 puntos – 15 min.) For each of the following system calls, give two conditions (**2 x 0,5 puntos**) that causes it to fail: fork, exec, and unlink.

Pregunta 2 (7 puntos – 35 min.) En la familia de los procesos creados por el siguiente programa algunos procesos respetan la Ley de De Morgan y otros no.

Explique el cálculo de la variable *r* en cada proceso (**3 puntos**),
 presente el árbol de procesos (**3 puntos**) e
 indique qué grupo de procesos acaba primero, qué grupo le sigue, etc. (**1 punto**).

```
$ cat deMorgan.c
#include <stdlib.h>
#include <stdio.h>

#define a fork()
#define b fork()

int
main(void)
{
    int r;

    printf("Shell PID = %d\n", getpid());
    r = ( !( a && b ) == (!a || !b) );
    while ( waitpid(-1, NULL, NULL) != -1);
    printf("%d->%d: %s\n", getpid(), getppid(), r?"True":"False");
    exit(r);
}

$ gcc deMorgan.c -o deMorgan

$ ./deMorgan
...
```

Pregunta 3 (7 puntos – 35 min.) Como se sabe, en los sistemas Unix todo son archivos, y cada archivo tiene su *inode* único. El valor de *inode* es parte de la estructura *stat* que se obtiene con la función *fstat* indicando el descriptor del archivo en interés. En el siguiente programa se imprimen algunos *inodes*.

Su tarea es entender el código del programa y presentar

el árbol de los procesos (**1,5 puntos**),

las comunicaciones entre los procesos (**1,5 puntos**) y

la salida que produce el programa con unos valores arbitrarios pero relacionados (**3 puntos**).

Si la salida se presenta en el orden correcto, usted obtiene **1 punto** más.

```
$ cat pipes_inodes.c
#include <sys/stat.h>
#include <stdlib.h>
#include <stdio.h>

#define n 3

int
main(void)
{
    int fd[2], i=0, p;
    struct stat sb;

    pipe(fd); dup2(fd[0],0); dup2(fd[1],1); close(fd[0]); close(fd[1]);
    while (i < n) {
        pipe(fd);
        p=fork();
        p ? dup2(fd[1],1) : dup2(fd[0],0);
        close(fd[0]); close(fd[1]);
        if (p) break;
        i++;
    }
    while ( waitpid(-1, NULL, NULL) != -1 );
    fprintf(stderr, "Process %d. inodes: ", getpid());
    fstat(0, &sb);
    fprintf(stderr, "0(%ld), ", sb.st_ino);
    fstat(1, &sb);
    fprintf(stderr, "1(%ld)\n", sb.st_ino);
    exit(0);
}
$ gcc pipes_inodes.c -o pipes_inodes
$ ./pipes_inodes
...
```

Pregunta 4 (2 puntos – 10 min.) Las variables compartidas entre 2 procesos concurrentes (o hilos) *x*, *y* tiene sus valores iniciales 0. Un proceso ejecuta la sentencia $x = y + 1$ en el mismo tiempo que otro proceso ejecuta la sentencia $y = x + 1$. ¿Cuáles serán los valores finales de las variables?



La práctica ha sido preparada por VK
con LibreOffice Writer.

Profesor del curso: (0781) V. Khlebnikov

Pando, 15 de abril de 2014