

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

1ra práctica (tipo a)
(Primer semestre de 2013)

Horario 0781: prof. V.Khlebnikov

Horario 0782: prof. A.Bello R.

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 (3 puntos) Responda a las siguientes preguntas:

- a) **(0,5 puntos)** ¿Cómo se llama el programa que interactúa con los usuarios, que está basado en texto, que no es parte del sistema operativo pero que lo usa para conseguir su trabajo?
- b) **(0,5 puntos)** ¿Cuáles son las formas en que se pueden multiplexar los recursos?
- c) **(0,5 puntos)** UNIX tuvo dos grandes variantes de desarrollo ¿Cuáles fueron estas?
- d) **(0,5 puntos)** ¿Cómo se llamaba el sistema operativo que dominaba el mundo de las microcomputadoras antes que IBM lanzara su IBM PC?
- e) **(0,5 puntos)** En la línea de comando de una terminal ¿que se espera que se redireccione con la siguiente orden: `find / -name '*.c' > salida 2>&1`?
- f) **(0,5 puntos)** La función de librería `opendir()` abre un directorio y devuelve un puntero que luego permite leer las entradas del directorio ¿Por qué `opendir()` no se diseñó como una llamada al sistema?

Pregunta 2 (8 puntos) Analice el siguiente programa:

```
/* 2013-1 Practica 1, vk */
```

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

```
int
main(void)
{
    int i;
```

```

int MiB = 1024*1024;
int pfd1[2], pfd2[2] ;
char bs[8];
int bytes_read, total_bytes_read;
char buffer[1024];

for (i=1; i<=MiB; i<=1) { /* bitwise left shift: 1,2,4,8,...,1048576 */
    if ( waitpid(fork(),(int *)NULL,0) == -1 ) {
        if ( !pipe(pfd1) && !pipe(pfd2) && fork() ) {
            close(pfd1[1]); close(pfd2[1]); total_bytes_read=0;
            while ( (bytes_read=read(pfd1[0],buffer,i)) != 0 )
                total_bytes_read += bytes_read;
            while ( (read(pfd2[0],buffer,MiB)) != 0 )
                printf("%s\n", buffer);
            printf("total %d bytes read\n-----\n\n", total_bytes_read);
            exit(0);
        } else {
            close(pfd1[0]); dup2(pfd1[1],1); close(pfd1[1]);
            close(pfd2[0]); dup2(pfd2[1],2); close(pfd2[1]);
            sprintf(bs,"bs=%d",i);
            execl("/usr/bin/time", "time", "dd", "if=/dev/zero", bs, "count=1024", (char *)NULL);
            abort();
        }
    }
}
exit(0);
}

```

Los resultados de ejecución de este programa son los siguientes:

```

$ gcc 2013-1_pr1.c -o 2013-1_pr1
$ ./2013-1_pr1
1024+0 records in
1024+0 records out
1024 bytes (1.0 kB) copied, 0.00273996 s, 374 kB/s
0.00user 0.00system 0:00.00elapsed 100%CPU (0avgtext+0avgdata 872maxresident)k
0inputs+0outputs (0major+272minor)pagefaults 0swaps

```

```

total 1024 bytes read
-----

```

```

1024+0 records in
1024+0 records out
2048 bytes (2.0 kB) copied, 0.0037624 s, 544 kB/s
0.00user 0.00system 0:00.00elapsed 80%CPU (0avgtext+0avgdata 868maxresident)k
0inputs+0outputs (0major+271minor)pagefaults 0swaps

```

```

total 2048 bytes read
-----

```

```

1024+0 records in
1024+0 records out
4096 bytes (4.1 kB) copied, 0.00404727 s, 1.0 MB/s
0.00user 0.00system 0:00.00elapsed 66%CPU (0avgtext+0avgdata 872maxresident)k
0inputs+0outputs (0major+272minor)pagefaults 0swaps

```

```

total 4096 bytes read
-----

...

1024+0 records in
1024+0 records out
536870912 bytes (537 MB) copied, 0.979887 s, 548 MB/s
0.00user 0.67system 0:00.98elapsed 69%CPU (0avgtext+0avgdata 1376maxresident)k
0inputs+0outputs (0major+398minor)pagefaults 0swaps

total 536870912 bytes read
-----

1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 1.52079 s, 706 MB/s
0.00user 1.05system 0:01.52elapsed 69%CPU (0avgtext+0avgdata 1888maxresident)k
0inputs+0outputs (0major+526minor)pagefaults 0swaps

total 1073741824 bytes read
-----

$

```

- a) **(2 puntos)** Deduzca qué devuelve `waitpid` en cada caso y cómo funcionará `if` de esta línea.
- b) **(2 puntos)** ¿Cómo funcionará la línea con doble `pipe` y un `fork`?
- c) **(2 puntos)** Explique las actividades en que están involucrados cada uno de los procesos. El manual de `time` dice que "`time - run programs and summarize system resource usage: time COMMAND [ARGS]`". El manual de `dd` dice que "`dd - copy a file, converting and formatting according to the operands: dd [OPERAND]...`", donde el operando `if=` indica el archivo de fuente (input file), el operando `bs=` indica el tamaño del bloque, el operando `count=` indica la cantidad de bloques a copiar.
- d) **(2 puntos)** Indique exactamente qué información pasa por cada tubo (y de dónde proviene). Y cuál es el objetivo del programa dado.

Pregunta 3 (4 puntos)

- a) **(2 puntos)** Explique con un ejemplo la razón que justifique la siguiente línea: "To prevent endless recursion, most modern operating systems do not allow hard links on directories"
- b) **(2 puntos)** ¿Qué diferencia hay entre las llamadas al sistema `stat()` y `lstat()` cuando ambas se aplican a un enlace simbólico?

Pregunta 4 (2 puntos)

- a) **(1 punto)** [From OSC7E] “As in all cases of modular design, designing an operating system in a modular way has several advantages. The system is easier to debug and modify because changes affect only limited sections of the system rather than touching all sections of the operating system. Information is kept only where it is needed and is accessible only within a defined and restricted area, so any bugs affecting that data must be limited to a specific module.” ¿De qué tipo de estructura del sistema operativo se habla aquí?
- b) **(1 punto)** Indique los ejemplos de sistemas operativos que corresponden a la siguiente descripción: [From OSC7E] “Benefits typically include the following (a) adding a new service does not require modifying the kernel, (b) it is more secure [...], and (c) [the] [...] design and functionality typically results in a more reliable operating system.”

Pregunta 5 (3 puntos) Enumere cada uno de los requisitos para la exclusión mutua.



La práctica fue preparada por AB(1,3,5) y VK(2,4)

Profesores del curso: V.Khlebnikov
A.Bello R.

Pando, 9 de abril de 2013