

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

1ra práctica (tipo a)
(Primer semestre de 2012)

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) ¿Cuál es la diferencia sustancial entre `printf` y `write`?
- b) (0,5 puntos) La memoria de los procesos en UNIX se divide en tres segmentos. ¿Cuáles son estos?
- c) (0,5 puntos) ¿Qué significa el código de protección de archivos en UNIX igual a 0755?
- d) (0,5 puntos) ¿Qué es un archivo especial de caracteres (*character special file*)?
- e) (0,5 puntos) ¿Qué contiene *Basic Input Output System* y cuál es su objetivo?
- f) (0,5 puntos) ¿Qué es *X Window System*?

Pregunta 2 (7 puntos) El siguiente programa verifica si (fork - wait) implica a (fork + exec):

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

int
main(int argc, char **argv)
{
    char p[8] = "/bin/ps";
    char a0[3] = "ps";
    char a1[2] = "l";

    printf("(fork - wait) -> (fork + exec) = !(fork - wait) OR (fork + exec)\n");
    fflush((FILE *)NULL); /* all open output streams */
    if ( !(fork() - wait((int *)NULL)) || (fork() + execl(p,a0,a1,(char *)NULL)) ) {
        sleep(1); printf("\nYes!\n"); exit(0); }
    printf("\nNo...\n");
    exit(1);
}
```

donde el manual de wait dice que:

`wait()`

The `wait()` system call suspends execution of the calling process until one of its children terminates.

RETURN VALUE: on success, returns the process ID of the terminated child; on error, -1 is returned.

- a) (1 punto) Presente el árbol de los procesos creados por este programa.
- b) ((6/n) puntos) Describa el hilo de ejecución y los resultados que se producen durante la vida (del inicio al fin) de cada proceso creado por el programa.

Pregunta 3 (7 puntos) (*UnixTM Systems Programming: Communication, Concurrency, and Threads* - Kay A. Robbins, Steven Robbins)

```
cat ring.c
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
int main(int argc, char *argv[ ]) {
    char mensaje[20];
    pid_t childpid;          /* indicates process should spawn another */
    int error;               /* return value from dup2 call */
    int fd[2];              /* file descriptors returned by pipe */
    int i;                  /* number of this process (starting with 1) */
    int nprocs;             /* total number of processes in ring */
    /* check command line for a valid number of processes to generate */
    if ( (argc != 2) || ((nprocs = atoi (argv[1])) <= 0) ) {
        fprintf (stderr, "Usage: %s nprocs\n", argv[0]);
        return 1;
    }
    if (pipe (fd) == -1) {    /* connect std input to std output via a pipe */
        perror("Failed to create starting pipe");
        return 1;
    }
    if ((dup2(fd[0], STDIN_FILENO) == -1) ||
        (dup2(fd[1], STDOUT_FILENO) == -1)) {
        perror("Failed to connect pipe");
        return 1;
    }
    if ((close(fd[0]) == -1) || (close(fd[1]) == -1)) {
        perror("Failed to close extra descriptors");
        return 1;
    }
    for (i = 1; i < nprocs; i++) {    /* create the remaining processes */
        if (pipe (fd) == -1) {
            fprintf(stderr, "[%ld]:failed to create pipe %d: %s\n",
                (long)getpid(), i, strerror(errno));
            return 1;
        }
    }
}
```

```

if ((childpid = fork()) == -1) {
    fprintf(stderr, "[%ld]:failed to create child %d: %s\n",
            (long)getpid(), i, strerror(errno));
    return 1;
}
if (childpid > 0) /* for parent process, reassign stdout */
    error = dup2(fd[1], STDOUT_FILENO);
else /* for child process, reassign stdin */
    error = dup2(fd[0], STDIN_FILENO);
if (error == -1) {
    fprintf(stderr, "[%ld]:failed to dup pipes for iteration %d: %s\n",
            (long)getpid(), i, strerror(errno));
    return 1;
}
if ((close(fd[0]) == -1) || (close(fd[1]) == -1)) {
    fprintf(stderr, "[%ld]:failed to close extra descriptors %d: %s\n",
            (long)getpid(), i, strerror(errno));
    return 1;
}
if(childpid)
    break;
}
return 0;
}

```

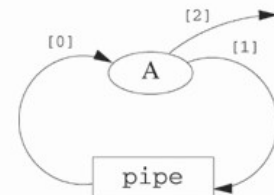
- a) (2 puntos) Elabore un diagrama mostrando el árbol de procesos y las conexiones hechas por las tuberías cuando el programa se ejecuta como `./ring 4`. El diagrama debe ser de forma análoga a como se muestra a continuación. Por ejemplo al terminar de ejecutar el siguiente código:

```

int fd[2];
pipe(fd);
dup2(fd[0], STDIN_FILENO);
dup2(fd[1], STDOUT_FILENO);
close(fd[0]);
close(fd[1]);

```

Se tiene el siguiente diagrama:



- b) (2 puntos) Escriba las líneas necesarias para que el proceso padre envíe un mensaje y éste mensaje pueda recorrer a través de toda la conexión formada. Además indique en dónde se debería insertar dicho código.
- c) (3 puntos) Modifique el programa para que lleve a cabo el equivalente a la siguiente orden de *shell*: `ps | sort | lpr`. Indique cómo debe ser invocado el programa ejecutable.

Pregunta 4 (3 puntos)

- a) (0,5 puntos) En UNIX la llamada al sistema que permite eliminar un proceso es ... (¿?)
- b) (0,5 puntos) Considere un ingeniero informático con aficiones culinarias que está horneando un pastel de cumpleaños para su hija; tiene una receta del pastel y una cocina bien surtida con todos los materiales: harina, huevos, azúcar, extracto de vainilla, etc. En esta analogía, ¿quién es el programa y quién es el proceso?
- c) (0,5 puntos) ¿A qué procesos se les denomina **demonios** (*daemons*)?

- d) **(0,5 puntos)** De manera individual, cuando un proceso recibe una señal, ¿qué puede hacer el proceso con la señal?
- e) **(0,5 puntos)** Se ha calculado que el tiempo de CPU de un proceso es $5u$ y el de su E/S es de $3u$ (se ha incluido los tiempos de cambios de contexto). Si es el único proceso en la cola de *listos*, ¿cuál es el motivo para que este proceso tenga un tiempo total de más de $8u$?
- f) **(0,5 puntos)** ¿Cómo se llama la estructura que contiene la dirección del procedimiento del servicio de interrupción?



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

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

Pando, 4 de abril de 2012