

CC31A Examen

2 horas

Noviembre de 2006

Pregunta 1 (Conceptos)

- (C)

En un programa en C, debemos mantener en memoria una tabla grande que es útil para algunas funciones del programa. Explique las diversas alternativas que C provee para su almacenamiento en diversas partes de la memoria y discuta cómo decidiría el lugar donde almacenar esta tabla.

- (longjmp)

Tengo la siguiente función:

```
jmp_buf err;
void f() {
    setjmp(err);
    return;
}

main()
{
    f();
    ....
    longjmp(err, 1);
    ...
}
```

El programa llama a `f()` y luego hace un `longjmp` y el programa se cae. ¿Porqué?

- (señales)

¿Qué cree Ud que debiera pasar si un programa que usa múltiples threads recibe una señal? Analice las alternativas y discuta cual es más razonable que otra y porqué.

- (procesos)
En un sistema Linux, nos encontramos con que existen muchos procesos del servidor web que están en estado Zombie. ¿Qué puede estar pasando? ¿Es grave?
- (threads)
Explique qué es un *busy wait* y porqué hay que evitarlo.
- (perl)
¿Ud usaría perl para programar algunas aplicaciones? ¿Por qué? (está prohibido decir "por que no me gusta").

Pregunta 2

Escriba el código de la función `read_cmd` que ejecuta un comando shell y entrega un filedescriptor desde donde leer su salida estándar:

```
int read_cmd(char *cmd);

/* Ejemplo de uso */
main() {
    int fd;

    fd = read_cmd("ls /usr");
    if(fd < 0) error...

    while((cnt=read(fd, buf, BUF_SIZ)) > 0)
... procesa buf de tamaño cnt
}
```

Pregunta 3

Usando la función `read_cmd` de la Pregunta 2, escriba el código de un servidor de red que recibe un comando shell desde el cliente y luego lo ejecuta entregándole a través del socket la salida estándar del comando.

El servidor debe ser multi-clientes (atender a varios a la vez).

Pregunta 4

Un alumno propone este código como una buena solución para un buffer de tamaño 1 con múltiples productores y múltiples consumidores:

```
Init_sem(vacio, 1); Init_sem(lleno, 0);
```

```

char buffer;
Productor()
{
    char c;

    for(;;)
    {
        c = Producir();
        lock(mutex);
        P(vacio);
        buffer = c;
        V(lleno);
        unlock(mutex);
    }
}

Consumidor()
{
    char c;

    for(;;)
    {
        lock(mutex);
        P(lleno);
        c = buffer;
        V(vacio);
        unlock(mutex);
        Consumir(c);
    }
}

```

Evalúe la propuesta y proponga mejoras posibles.