

# CC3301 Programación de Software de Sistemas

Examen – Semestre Primavera 2012

Prof.: Luis Mateu

## Pregunta 1

Se define el tipo de datos Cola con las siguientes operaciones.

|                                 |  |
|---------------------------------|--|
| Cola *nueva_colas();            | Crea una nueva cola                                  |
| void agregar(Cola *c, char *s); | Agrega una copia del string s a la cola c            |
| char *extraer(Cola *c);         | Entrega y extrae el string más antiguo de la cola c  |
| int esta_vacia(Cola *c);        | Entrega 1 si la cola está vacía, 0 en caso contrario |

Defina la estructura de datos de Cola y programe las 4 operaciones.

**Restricciones:** Ud. debe usar listas simplemente enlazadas y el tiempo para agregar y extraer strings debe ser constante.

## Pregunta 2

Una isapre dispone de múltiples cajas de pago identificadas con un nombre. Cuando un afiliado (representado por un thread) necesita pagar invoca la función `solicitar_caja()` que entrega la identificación de una caja disponible. Si ninguna caja está disponible, entonces la función espera hasta que se desocupe alguna caja. Un empleado de la isapre (representado por otro thread) a cargo de por ejemplo la caja “C” indica cuando está disponible esa caja invocando la función `caja_disponible(“C”)`. Esta función retorna de inmediato. Una caja disponible se puede asignar a un y solo un afiliado y no se puede reasignar hasta que esté nuevamente disponible (lo que se indica con una nueva llamada de `caja_disponible`). Por simplicidad, no hay un orden definido para la atención de los afiliados.

Programe en C ambas funciones, más una función de inicialización. *Hint:* use la cola de la pregunta 1. Los encabezados son:

```
char *solicitar_caja();
void caja_disponible(char *id);
```

## Pregunta 3

Programe un servidor y un cliente para la asignación de cajas de una isapre por Internet. El servidor se llama `servidor_cajas`, corre siempre en `anakena` y escucha a los clientes en el puerto 3000. El cliente se llama `caja`. Un afiliado solicita una caja invocando `caja_s` y el empleado de la caja C notifica que la caja está disponible con `caja_d C`. El siguiente es un ejemplo de uso. Pedro, María y Diego son afiliados. Juan y Ana son empleados a cargo de cajas identificadas por sus nombres.

| Pedro                 | María                | Diego                | Juan               | Ana               |
|-----------------------|----------------------|----------------------|--------------------|-------------------|
|                       |                      |                      | % caja d Juan<br>% |                   |
| % caja s<br>Juan<br>% |                      |                      |                    |                   |
|                       | % caja s<br>(espera) |                      |                    |                   |
|                       | Ana<br>%             |                      |                    | % caja d Ana<br>% |
|                       |                      | % caja s<br>(espera) |                    |                   |
| % caja s<br>(espera)  |                      |                      |                    |                   |
| Ana<br>%              |                      |                      |                    | % caja d Ana<br>% |
|                       |                      | Juan<br>%            | % caja d Juan<br>% |                   |

*Hint:* use en el servidor las funciones de la pregunta 2.

## Pregunta 4

i. ¿Cuál es la diferencia entre una función de biblioteca y una llamada al sistema? ¿A qué categoría pertenece `malloc`? ¿Y `fork`? Dé otros 2 ejemplos de funciones de biblioteca y otros 2 ejemplos de llamadas al sistema.

ii. Discuta si esta paralelización de quicksort funciona o no:

```
void quicksort(int a[], int i, int j) {
    if (i<j) {
        int h= particionar(a, i, j);
        pid_t pid= fork();
        if (pid==0) {
            quicksort(a, i, h-1);
            exit(0);
        }
        else {
            quicksort(a, h+1, j);
            waitpid(pid, NULL, 0);
        }
    }
}
```

iii. Un proceso almacena en el arreglo `fds`, los identificadores de N archivos abiertos con `open`. Se necesita leer de algún archivo en donde hayan datos para leer. ¿Qué función usaría y por qué?

iv. ¿Qué hace el siguiente programa en Perl?

```
while ( $linea=<STDIN> ) {
    if ( $linea =~ /$ARGV[0]/ ) {
        print $linea;
    }
}
```

Dé un ejemplo de entrada del programa y su salida.