CC3301 Programación de Software de Sistemas – Examen – Semestre Otoño 2016 – Prof.: Luis Mateu

Pregunta 1

Parte a.- Programe la función *borrarBits* declarada como:

```
typedef unsigned long ulong;
ulong borrarBits(ulong x, ulong p, int n);
```

Esta función entrega el resultado de reemplazar por ceros la primera aparición del patrón *p* de *n* bits en el número *x*. Si el patrón no aparece se entrega *x*. En los siguientes ejemplos de uso la notación *0b...* expresa números en base 2 para facilitar la comprensión (aunque no sea parte del lenguaje C).

```
ulong r;
r= borrarBits(0b10110011, 0b110, 3);  // r es 0b10000011
r= borrarBits(0b11100000, 0b1, 1);    // r es 0b11000000
r= borrarBits(0b1111011, 0b1001, 4); // r es 0b1111011
```

Restricción: Ud. no puede usar los operadores de multiplicación, división o módulo (* / %). Use los operadores de bits.

Parte b.- Programe la siguiente función:

```
void elimEspacios(char *s);
```

Esta función reemplaza múltiples espacios contiguos en el string *s* por un solo espacio. Ejemplo de uso:

```
char s[80];
strcpy(s, "hola     que     tal");
elimEspacios(s); // s es "hola que tal"
```

Restricción: Ud. no puede usar el operador de subindicación [ ], ni su equivalente **(p+i)*. Para recorrer el string use el operador ++. Use múltiples punteros para direccionar distintas partes del string. No olvide terminar *s*.

Pregunta 2

I. Programe la función *mezclar* que une 2 listas enlazadas ordenadas ascendentemente. Su declaración es:

```
typedef struct nodo {
  int x;
  struct nodo *prox;
} Nodo;

Nodo *mezclar(Nodo *h1, Nodo *h2);
```

El siguiente ejemplo de uso muestra el resultado de invocar *h=mezclar(h1, h2)*. Los punteros *h1*, *h2* y *h* son de tipo *Nodo**.

Restricciones: Ud. no puede usar ninguna forma de iteración (while, for, etc.). Ud. debe usar recursividad (¡la versión no recursiva es complicada!). Ud. no puede usar *malloc*. Reutilice los nodos de los argumentos.

II. Ud. dispone de la función *integral* que calcula la integral de una función:

```
typedef double (*Funcion)(void *ptr, double x);
double integral(Funcion f, void *ptr,
                double xi, double xf, int n);
```

La función $integral(f, ptr, xi, xf, n)$ calcula $\int_{xi}^{xf} f(ptr, x)\,dx$ numéricamente. El parámetro *n* corresponde al número de subintervalos que se usan para estimar la integral. Programe la función *integral_par* que calcula la integral en paralelo usando 2 cores. Su encabezado es:

```
double integral_par(Funcion f, void *ptr,
                    double xi, double xf, int n);
```

Restricción: Ud. debe usar *fork* para crear un proceso pesado (*no threads*). El padre y el hijo calculan la integral en paralelo usando la mitad de los subintervalos cada uno. El hijo entrega su resultado al padre usando un *pipe*.

Pregunta 3

Aparte de comer y pensar, los 5 filósofos también necesitan ir al baño. El baño del restaurant es pequeño y por lo tanto puede ser usado por un solo filósofo a la vez. En la solución de más abajo los filósofos son representados por 5 threads que ejecutan la función *filosofo*. Esta solución es incorrecta porque no asegura la exclusión mutua de los filósofos al ocupar el baño.

```
void filosofo(int i) { // código para el filosofo i
  for (;;) {
    comerYPensar(i);    // función dada
    ocuparBano();       // función dada
} }
```

Corrija este código de manera que sí se asegure la exclusión mutua de los filósofos al ocupar el baño. Además los filósofos ancianos (filósofos 0 y 1) deben tener prioridad por sobre los filósofos jóvenes (2, 3 y 4). Esto significa que cuando un filósofo desocupa el baño, si hay filósofos ancianos esperando, los ancianos deben tener prioridad por sobre los jóvenes para ingresar al baño.

Ayuda: Para resolver este problema Ud. necesitará usar variables globales como un mutex, una o más condiciones, una variable que indique si el baño está ocupado o no y un contador para el número de ancianos en espera. Mantenga el mutex libre (*unlocked*) mientras se ocupa el baño. Si los 2 ancianos esperan el baño, se atienden en cualquier orden. Lo mismo para los jóvenes. Ignore el problema de la exclusión mutua de los palitos.