

CC3301 Programación de Software de Sistemas – Control 2

Semestre Otoño 2017 – Prof.: Luis Mateu

Pregunta 1

El cuadro de la izquierda muestra el contenido de un archivo que representa un diccionario no ordenado:

| | | | |
|-----------------|---|--------------|---|
| casa | edificacion construida para ser habitada | casa | edificacion construida para ser habitada |
| lluvia | condensacion del vapor de agua contenida en las nubes | lluvia | condensacion del vapor de agua contenida en las nubes |
| alimento | sustancia ingerida por un ser vivo | radio | receptor de ondas electromagneticas para audio |
| celular | aparato portatil de un sistema de telefonía celular | celular | aparato portatil de un sistema de telefonía celular |
| gato | felido domestico | gato | felido domestico |
| radio | receptor de ondas electromagneticas para audio | | |

Cada línea del archivo es de 80 caracteres y almacena una definición. Los primeros 10 caracteres corresponden a la palabra definida, luego vienen 69 caracteres para su definición y se termina la línea con un $\backslash n$ para completar los 80 caracteres. El archivo podría contener 0 o más líneas en blanco al final (y solo al final).

Programe la función `void eliminar(char *arch, char *pal)` que elimina la palabra `pal` del diccionario almacenado en el archivo de nombre `arch`. Para eliminar la palabra Ud. debe mover la última definición del diccionario hacia la posición de la palabra eliminada. En el ejemplo el diccionario de la derecha es el resultado de eliminar la palabra “alimento” del diccionario de la izquierda.

Restricción: El diccionario no cabe en la memoria. Use `fseek` astutamente para reducir la lectura/escritura del archivo. Sin embargo, la búsqueda de la palabra debe ser secuencial. Tenga cuidado en considerar que `strcmp("casa", "casa ")` es distinto de 0.

A modo de recuerdo, esta es la API para manejar archivos:

```
FILE *fopen(char *nom_arch, char *modo); // modo es "r+" para lectura/escritura
size_t fread(char *buf, size_t ancho_item, int n_item, FILE *file);
size_t fwrite(char *buf, size_t ancho_item, int n_item, FILE *file);
int fseek(FILE *file, long displ, int w); // w==SEEK_SET
int fclose(FILE *file);
```

Pregunta 2

Parte a.- (4 puntos) Se necesita estimar el mínimo de una función $f(x)$ en un intervalo $[x_0, x_1]$. Para ello se debe evaluar esta función en $m*p+1$ valores de x . Específicamente se debe evaluar $f(x_0+i*dx)$ con $i=0, \dots, m*p$, con $dx=(x_1-x_0)/(m*p)$ y retornar el mínimo de todas estas evaluaciones. La evaluación de la función f es costosa en tiempo de CPU y por ello se necesita paralelizar.

Programe la siguiente función:

```
typedef double (*Funcion)(double x);
double min_par(Funcion f, int p, double x0, double x1, int m);
```

La función `min_par` debe crear p threads para estimar el mínimo de f en paralelo. Cada thread calcula el mínimo de m evaluaciones de f . En total se harán $m*p+1$ evaluaciones de f . El thread en donde se invoca `min_par` evalúa f en x_1 y retorna el mínimo de todas las evaluaciones de f .

Parte b.- (2 puntos) En cátedra se resolvió el problema de los lectores/escritores. Los lectores consultan una estructura y los escritores la modifican. Se requiere que los lectores hagan sus consultas en paralelo mientras que los escritores deben modificar la estructura en exclusión mutua con otros lectores o escritores. Los lectores solicitan permiso para iniciar una consulta llamando a `enterRead()` y notifican el término de la consulta llamando a `exitRead()`. El protocolo de los escritores es análogo llamando a `enterWrite()` y `exitWrite()`. Estas funciones no reciben parámetros y no retornan nada. Estas 4 funciones se implementaron de modo que (i) un lector no puede iniciar una consulta si hay un escritor modificando, y (ii) un escritor tampoco puede iniciar una modificación si ya hay lectores consultando u otro escritor modificando.

Reprograme estas 4 funciones de modo que le den prioridad a los escritores por sobre los lectores. Para lograrlo se agrega la restricción (iii): ningún lector puede iniciar una consulta mientras hayan escritores esperando iniciar una modificación.