UNIVERSIDAD DE COSTA RICA

Escuela de Ciencias de la Computación e Informática

CI-0117 Programación paralela y concurrente

Grupos 1 y 2

Enunciado tarea programada II

Fecha de entrega: 2020/Jun/12

Modalidad: **individual**

Contador de etiquetas HTML

El lenguaje HTML está caracterizado por emplear etiquetas o marcas que indican las características del documento que integran. Cada marca HTML está contenida en los separadores "<" y ">", los cuales no forman parte de la etiqueta. Las etiquetas son interpretadas para poder presentar al usuario el documento en el formato original pensado por su creador.

En esta tarea, deben construir un programa en C++ para contar todas las etiquetas que aparecen en un conjunto de archivos HTML que se pasará como parámetros. El programa debe contar todas las etiquetas contenidas en los archivos indicados y mostrar un listado de esas etiquetas y su cantidad de apariciones, en orden alfabético de menor a mayor. Además, el usuario puede indicar, como parámetro también, la cantidad de trabajadores con que pretende realizar el trabajo y la estrategia de asignación de líneas a los trabajadores.

Se pretende que cada archivo HTML sea procesado por un conjunto de trabajadores independientes ("contadores"), este es el parámetro que el usuario puede indicar. Cada archivo HTML tendrá entonces un programa "lector" que se encargará de crear una instancia de la clase lectora de archivos (*FileReader*) con su respectiva estrategia y el nombre de archivo HTML que le toca trabajar, también generar la cantidad de trabajadores indicados por el usuario para dividir el archivo en grupos de líneas y procesarlas independiente. Por ejemplo, si el usuario indica tres archivos HTML, "uno.html", "dos.html" y "tres.html" e indica que quiere utilizar 10 trabajadores y las estrategias de lectura, entonces, vamos a tener un trabajador "html-maestro" que se encarga de crear los procesos necesarios para contar los archivos HTML indicados, en este ejemplo 3, además vamos a tener tres "maestro-lector" que se estarán a cargo de crear la instancia de la clase lectora con la estrategia indicada y de generar los 10 trabajadores solicitados por el usuario, completando en total 34 trabajadores para contar las etiquetas HTML de este ejemplo. Cada "maestro-lector" debe generar su listado de etiquetas para comunicarlo al "super-maestro", quien debe acumular el total de todas la etiquetas de todos los archivos y presentar la salida al usuario.

Debe construir una clase C++ (*FileReader*) que sea capaz de procesar un archivo de texto HTML, con ***H*** líneas, siguiendo varias estrategias de acuerdo con la cantidad de trabajadores (***t***) que el usuario pretenda utilizar. Esta clase será la única encargada de manipular las lecturas del archivo y almacenar no más de 512 bytes del archivo HTML.

Las estrategias serán similares a las que hemos revisado en las clases:

- Dividir el total de líneas del archivo HTML entre los ***t*** trabajadores y entregar una porción a cada trabajador (***H***/***t***)

- Entregar al primer trabajador (0) todas las líneas cuyo resto de dividir el número de línea entre ***t*** sea 0; entregar al segundo trabajador todas las líneas cuyo resto de dividir el número de línea entre ***t*** sea 1 y así sucesivamente. Note que en estas dos estrategias la cantidad de líneas asignadas a cada trabajador es aproximadamente la misma

- Entregar una línea del archivo a cada trabajador por demanda, cada vez que un trabajador requiere una línea le es entregada la siguiente línea disponible del archivo, el lector debe saltar a la siguiente línea. Debe sincronizar los trabajadores para que dos de ellos no reciban la misma línea y el conteo de etiquetas no sea correcto

- Cada estudiante debe programar una estrategia adicional de su propia invención y que no replique las estrategias indicadas en este enunciado, puede utilizar la teoría revisada de OpenMP para planear su estrategia

El constructor de esta clase recibirá el nombre del archivo a procesar, la cantidad de trabajadores que pretenden utilizar ese archivo y la estrategia a utilizar. Debe proveer métodos que faciliten la lectura del archivo por parte de los trabajadores, por ejemplo “hasNext()” y “getNext()” de manera que cada trabajador pueda manipular el archivo en una estructura similar a la siguiente:

```
void contarEtiquetas( Lectores lector, int worker ) {
   char buffer[ 512 ];
   while ( lector→hasNext( worker ) ) {
      buffer = lector→getNext( worker, &buffer );
      // Contar las etiquetas
   }
}
```

Esquema

Suponiendo que el usuario llama a su programa de la siguiente manera:

```
contar –t=10 a.html b.html c.html –e=2,3,4
```

Su programa debe crear tres trabajadores “lectores” para procesar un archivo HTML cada uno, el primero “a.html”, el segundo “b.html” y el tercero “c.html”. Todos estos “lectores” a su vez, deben crear otros trabajadores “contadores” (10) para contar la cantidad de veces que aparece una determinada etiqueta en las porciones del archivo que le tocó a cada uno, siguiendo la estrategia

indicada, el primero la segunda estrategia (parámetro *-e*), el segundo la tercera y el tercero la cuarta. Una vez que estos “contadores” tienen las etiquetas contadas, deben pasar ese información a su “lector”, quien acumulará toda la información (etiquetas, veces) de todos sus “contadores” y enviará esa información al trabajador original, el cual a su vez acumulará la información de todos los archivos HTML indicados por el usuario.

Restricciones adicionales

- Los trabajadores solo podrán tener en memoria una línea del archivo a la vez o un buffer de trabajo de 512 bytes. Los trabajadores no están autorizados a cargar toda la porción de su archivo en memoria de una sola vez
- La estrategia de lectura del archivo HTML no se puede cambiar una vez que el archivo ha sido abierto
- El programa debe poder correrse de manera serial, es decir, un solo trabajador que procese todo el archivo HTML. No importa si necesita crear los trabajadores intermediarios
- Los errores deben controlarse adecuadamente, por ejemplo, archivos que no existen, estrategias inválidas, trabajadores negativos, etc.
- Puede utilizar la estructura que considere necesaria para almacenar las etiquetas HTML y la cantidad de veces que aparecen, lo que no es permitido hacer es pasar esa estructura completa o referencias entre los distintos trabajadores
- Puede comunicar los trabajadores utilizando cualquiera de los métodos que hemos revisado en el curso
- Puede utilizar un archivo de configuración para establecer los parámetros que el usuario quiere pasar al programa, en lugar de tener que escribirlos cada vez que el usuario quiera usar el programa
- El programa debe medir los tiempos empleados para procesarlos correctamente, de manera que sea posible determinar cuál fue la mejora por utilizar varios trabajadores

Referencias

Etiquetas HTML: https://way2tutorial.com/html/tag/index.php