

Reflexión

En esta actividad se nos pidió crear una aplicación que leyera el archivo “bitacoraACT5_2.txt”, almacenara la información en el dentro de una tabla hash y, dado una dirección IP, despliegue la información obtenida de la bitácora perteneciente a esa IP.

Empezamos la aplicación pensando en como podríamos almacenar la información, por cuestiones de eficiencia utilizamos la estructura de una tabla hash que en C++ se puede implementar mediante un unordered map o mapa desordenado. En el unordered map utilizamos la string de la IP como key para acceder a los datos, y un par en el que el primer elemento es un entero que representa la cantidad de veces que se accede a una IP mientras que su segundo elemento es un par que contiene dos vectores de strings, el primero almacenando las fechas de intentos de acceso y el segundo almacenando los motivos de los intentos de acceso.

Ya establecido como almacenaríamos los datos empezamos a formar la función leerArchivo, que recibiría por referencia la tabla hash que llamamos como variable hashtable. En esta función se lee la bitácora y se almacenan los datos de esta en sus respectivas variables, leyendo el mes, el día, la hora, la dirección IP y el motivo de cada registro de intento de acceso que se encuentra en la bitácora. Las variables de mes, día y hora se almacenan en una variable fecha y se elimina el puerto en la dirección IP.

Para almacenar esta información en la hashtable se usa la IP como key, se revisa si los vectores de datos están vacíos, si lo están el conteo de accesos se establece como 1, de lo contrario se le suma uno. Posteriormente se accede a los vectores y se les añade los nuevos elementos fecha y motivo. Finalmente se cierra el archivo de la bitácora para evitar daños. Con esto se cumplen los primeros dos requerimientos de la actividad con una complejidad de $O(n)$, aunque considerando las complejidades indefinidas de los métodos getline, subtr y find podría considerarse mayor.

Después formamos la función print que recibe un string de una IP y el hashtable. Con estos datos accede a los elementos almacenados en el nodo del unordered map con la clave de la IP y los despliega mostrando primero el número de accesos y después la fecha y el motivo de cada intento de acceso. Esta función tiene una complejidad de $O(n)$.

Esta actividad me pareció simple con la ayuda de mis compañeros de equipo y creo que logramos implementar adecuadamente distintos tipos de datos o estructuras como los son los unordered maps para un acceso más rápido a información.