

ALGORITMOS Y ESTRUCTURAS DE DATOS

Guía 7. LCD Fase II: Virtualización de Hardware y simulación con Twitter.

1. Se pide escribir un programa que reproduzca tuits de algún usuario público a elección (<https://developer.twitter.com/en/docs.html>) a través de los displays LCD virtuales que desarrollaron en el TP6.
 - a. Los tuits serán obtenidos de la web en formato JSON. El cual será interpretado según <https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user-timeline.html>. Para parsear archivos JSON se podrá utilizar la Nlohmann JSON (<https://github.com/nlohmann/json>) que explicaremos en clase.
 - b. El programa deberá consistir en una pequeña GUI creada con Dear ImGui que permita ingresar el nombre del usuario del que se quieren obtener los tuits y opcionalmente un número con la cantidad de tuits a bajar. Si no se recibiera este segundo parámetro se usará el default de Twitter.
 - c. Se deberá utilizar la librería libCurl (<https://curl.haxx.se/libcurl/>) para comunicarse con Twitter que ya explicamos en clase (TP5) ya que se necesita realizar una conexión HTTPS con la API de Twitter.
 - d. Si surgiera un error al intentar comunicarse con Twitter, o de la comunicación resultara que el usuario de Twitter especificado al correr el programa no existe o no tiene tuits, o se obtuviera cualquier otro error reportado por Twitter, se deberá indicar apropiadamente en el display explicándole al lector el problema encontrado.
 - e. Caso contrario, mientras se bajan los tuits del usuario, el programa deberá mostrar en la primera línea del LCD el usuario del que se están bajando los tuits (ej: @lanacion) en forma de marquesina si no entrara en los 16 caracteres de la pantalla, y en la segunda línea algún indicio de que está bajando los tuits; por ejemplo: la cantidad de tuits que lleva bajados, una barra de progreso o algún indicador dinámico (por ejemplo ciclar en el mismo casillero entre los caracteres '|', '\', '-', '/') para que el usuario entienda que el programa no se ha colgado. Se podrá cancelar en cualquier momento esta acción mediante la GUI (proveyendo alguna solución para tal fin, como un botón, por ejemplo). En dicho momento se terminará el proceso y se procederá a mostrar los tuits bajados hasta el momento (ignorando cualquier objeto JSON del arreglo que no se haya bajado en su totalidad); de lo contrario se esperará hasta bajar todos los tuits solicitados.
 - f. El programa debe reproducir todos los tuits bajados. Para saber cómo interpretar el JSON obtenido debemos revisar la documentación de Twitter: <https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user-timeline.html>.

- g. La forma de mostrar la información en el display es la siguiente:
 - g1. En la primera línea se mostrará la fecha y hora del tuit en el formato DD/MM/AA – hh:mm (totalizando 16 caracteres si se consideran los espacios que separan el carácter '-' entre de la fecha y la hora).
 - g2. En la segunda línea se mostrará el tuit empezando con el nombre del usuario que lo escribió como si fuera un diálogo. Por ejemplo si encontramos el siguiente titular de La Nación: *"Este era el cacerolazo de Coronel Diaz y Santa Fe hace un ratito"*, deberemos reproducir: *La Nacion: - Este era el cacerolazo de Coronel Diaz y Santa Fe hace un ratito -*
 - g3. El tuit que se muestra en la segunda línea deberá ser desplazado horizontalmente (término que en la jerga se conoce como *scrolleado*, del inglés *scroll*). Para ello las letras deberán desaparecer una a una por la margen izquierda al mismo tiempo que una a una cada letra nueva aparecerá por la margen derecha hasta completar el tuit rellenando las posiciones sobrantes del LCD con espacios. El desplazamiento deberá realizarse a un tiempo suficientemente lento como para que el lector pueda leer el tuit y a la vez suficientemente rápido como para que no se aburra.
 - g4. Se le debe permitir al usuario volver a mostrar el tuit actual, saltar al siguiente o uno hacia atrás. Además se le debe permitir salir del programa en cualquier momento y aumentar o disminuir la velocidad de desplazamiento del tuit. Toda estas acciones se debe poder comandar desde la GUI.
 - g5. Una vez mostrado el último tuit el programa informará en el display. "Ultimo Tuit". Desde la GUI se podrá elegir si se quieren mostrar más tweets o abandonar el programa.
- h. Entre tuits se debe incluir una pausa prudencia para que el lector pueda digerir la información que ha leído.
- i. Se le debe permitir al usuario poder alternar en tiempo real entre los distintos LCD, de forma de poder reproducir los tuits al mismo tiempo en uno o más LCDs distintos. Deberán proveer algún tipo de soporte para ello en la GUI que permita "crear" y "destruir" LCDs y al crearlos permitir elegir el tipo de LCD a crear. Cada grupo debería contar con tres tipos de LCD, uno por cada integrante realizados en el TP6.

Bibliografía

- Documentación de Twitter API: <https://developer.twitter.com/en/docs.html>
- Twitter API Authentication Docs:
<https://developer.twitter.com/en/docs/basics/authentication/overview/application-only>
- Spotify API Authentication Docs (excelente explicación del flujo de OAuth):
<https://developer.spotify.com/documentation/general/guides/authorization-guide/>
- Compilar libCurl en Visual 2017: <https://medium.com/@chuy.max/compile-libcurl-on-windows-with-visual-studio-2017-x64-and-ssl-winssl-cff41ac7971d>
- Funcion de libcurl para obtener que tamaño tiene lo que me devuelve el HTTP:
https://curl.haxx.se/libcurl/c/curl_easy_getinfo.html (en particular el campo CURLINFO_SIZE_DOWNLOAD_T).
- Interpretación de los tuits: https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.html