

El lenguaje de programación C permite al programador un control detallado de procesos, memoria y entrada/salida; además su diseño permite un mapeo muy cercano a instrucciones de máquina. Por estas razones se ha destacado como una plataforma para la eficiencia y ha encontrado un nicho en el campo de programación de sistemas: compiladores, intérpretes, sistemas operativos o sistemas empotrados, entre otros.

El objetivo de este proyecto es construir una aplicación dentro de este contexto. La aplicación a desarrollar será un sencillo servidor de HTTP con las siguientes características:

- El servidor escuchará únicamente *requests* con método GET. Cualquier otro tipo de request deberá responder con un código de error 501 (método no implementado).
- Cada *request* será procesado por un hilo independiente.
- El proceso de un *request* consistirá en *parsear* los encabezados HTTP recibidos para determinar cuál es el archivo que se está solicitando, leerlo y escribirlo al *response*. Si el archivo solicitado no se encuentra, el servidor responderá con un código 404 (no encontrado).

GET /docs/index.html HTTP/1.1

Host: localhost:8080

Connection: keep-alive

Cache-Control: max-age=0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)

Ubuntu Chromium/41.0.2272.76 Chrome/41.0.2272.76 Safari/537.36

Accept-Encoding: gzip, deflate, sdch

Accept-Language: en-US,en;q=0.8,es;q=0.6

- El servidor buscará los archivos en un directorio especial llamado *wwwroot*.
- El servidor se conectará a la red utilizando un *socket*. Cada vez que un request llegue al socket, el servidor lo leerá y lo colocará en una cola especial. Un *pool* de hilos estará consultando esta cola constantemente utilizando el patrón Productor/Consumidor. Sólo un hilo podrá leer de la cola a la vez.

- El puerto en el que socket escucha, el directorio wwwroot y el tamaño del pool de hilos deben ser argumentos de ejecución del servidor. El servidor debe ser ejecutado de la siguiente manera:

```
HTTPServer -R /home/diegomunguia/tmp/wwwroot -W 50 -P 8080
```

- El servidor debe funcionar correctamente cuando se accesa desde Firefox o Chromium.

Recursos

Programación con hilos

<http://www.cs.kent.edu/~ruttan/sysprog/lectures/multi-thread/multi-thread.html>

Cola productor/consumidor

<http://stackoverflow.com/questions/1212623/multiple-writer-thread-safe-queue-in-c>

Programación de sockets

https://en.wikipedia.org/wiki/Berkeley_sockets#Socket_API_functions

<http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html>

Logística

El proyecto debe ser trabajado individualmente. El proyecto debe ser entregado el día **martes 31 de marzo, 2015** a más tardar a las 11:59pm, la entrega se hará a través del TEC Digital. Se deberá subir a la carpeta de Proyectos del curso una carpeta que contenga el código fuente, la carpeta deberá estar nombrada con el nombre y apellido del estudiante **sin espacios** y el sufijo "-P1", ej: DiegoMunguia-P1.