

COMPUTACION II

TP3

Fecha de entrega Final: 29/09/2020

Problema

El objetivo del práctico es desarrollar un servidor web multi-proceso que pueda atender múltiples conexiones simultáneas.

Se debe especificar con la opción -p o --port el puerto donde espera conexiones nuevas. Con la opción -d o --document-root el directorio donde se encuentran los documentos web y con la opción -s --size la cantidad máxima de bytes que se irán leyendo de los documentos web.

En caso que no se solicite ningún documento, se debe responder con un archivo index.html de bienvenida.

También será posible aplicar filtros a archivos de imagen de tipo portátil pixmap (ppm). En ese caso se debe agregar al método GET solicitando el archivo una query indicando el filtro a solicitar. Los filtros disponibles son Red, Green, Blue y Black/White. A cada uno de estos filtros se les puede ajustar adicionalmente la escala de intensidad (1 no la modifica)

En caso se solicite un filtro, se debe implementar creando los hilos necesarios (en función de -s), quedando el hilo principal a la espera de que terminen los mismos. Se aclara que la respuesta con los filtros generados NO deben escribirse en el sistema de archivos, para evitar llenar el disco y disminuir la velocidad de respuesta con entrada/salida.

Requerimientos

- * La aplicación debe contener como mínimo 3 funciones.
- * Debe implementar el método de requerimiento GET al menos.
- * Debe devolver como mínimo, tres tipos de resultados al cliente: 200 Ok, 404 Not Found y 500 Internal Server Error (Ver especificación de HTTP 1.1) con los headers de respuesta Content-Length y Content-Type correspondientes.
- * Debe soportar archivos de tipo html, jpg, pdf y ppm.
- * La lectura de los archivos se debe hacer como máximo de a n bytes.
- * Debe procesar las opciones con getopt (agregar una opción de ayuda) o con argparse.
- * Debe soportar sockets de tipo IPv4 o IPv6 indistintamente.
- * Debe soportar al menos diez accesos concurrentes y mil requerimientos por acceso. (Apache benchmark - "ab -c 10 -n 1000" ... man ab)

Ejemplo modo de uso

```
~~~~~  
$ ./tp3.py -h  
usage: tp3.py [-h] -d DIR -p PUERTO -b SIZE
```

Tp3 - servidor web y filtro de ppm

optional arguments:

-h, --help	show this help message and exit
-d DIR, --documentroot DIR	Directorio donde estan los documentos web
-p PORT, --port PORT	Puerto en donde espera conexiones nuevas
-s SIZE, --size SIZE	Bloque de lectura máxima para los documentos

```
$ ./tp3.py --port 5000 -s 1024 -d /tmp
```

consultas:

```
wget http://192.168.2.2:5000/index.html
wget http://192.168.2.2:5000/imagen.jpg
wget http://192.168.2.2:5000/enunciado.pdf
wget http://192.168.2.2:5000/dog.ppm?filter=W&scale=1.2
```

Objetivos

- * Uso de mecanismos de IPC. Socket.
- * Manejo de concurrencia.
- * Manejo de archivos (apertura, escritura y cierre).
- * Creación de procesos e hilos.

Referencias

- * mime types en /etc/mime.types
- * RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0
- * Ver concepto de URI, URL, mime type, cabecera HTTP (request y response), HTTP status code.

Bonus Track

Manejar los errores o logging.

Generar un index.html con el contenido del directorio y un link a cada documento.