



## Micro servidor web

### Enunciado:

Empleando el lenguaje de programación C, se desea implementar un programa para implementar un servidor web bajo sistema operativo GNU/Linux. El mismo se ejecutará desde la consola y deberá quedar ejecutándose en segundo plano (*background*) hasta recibir una señal de finalización.

### Opciones del programa:

El programa implementado se deberá denominar: `servidorHTTP`, y deberá poder ejecutarse desde el *shell* en una terminal (consola). Las opciones posibles de invocación son:

```
servidorHTTP [IP][:puerto] [-h]
```

El significado de las diversas opciones del programa es el siguiente (recordar que las opciones que se presentan entre corchetes son opcionales):

- `[ip]`: número de IP al cuál se debe asociar el programa para escuchar los requerimientos que lleguen al puerto de escucha. Por defecto se debe asociar sólo al IP local del equipo (*localhost*).
- `[:puerto]`: número de puerto en el cuál se deben deprecionar las peticiones. Por defecto deberá ser el puerto 80 (protocolo *HTTP*).
- `[-h]`: Modo *ayuda* en línea: Si se especifica el parámetro `-h` en algún lugar de la línea de comandos, el programa sólo deberá mostrar una pequeña ayuda por pantalla, la cual consiste en listar las opciones de invocación del programa (sintaxis) y su semántica (con una aclaración sintética en base a la que se brinda en este enunciado --- no copiar, elaborar algo adecuado y como corresponde ---). En caso de estar presente esta opción no se deberá iniciar el servidor.

### Objetivo y funcionalidades:

Las funciones básicas del servidor web a implementar deben ser:

- Escuchar por peticiones de tipo GET, y responder en forma acorde con la información solicitada.
- Deberá analizar la petición y recuperar el archivo solicitado, pueden solicitarse archivos con extensiones: *HTM*, *HTML* y *PHP*.
- En el caso de solicitar un archivo con extensión *PHP* se deberán pasar los parámetros necesarios que vengan definidos en el método GET del solicitante.



# PROYECTO N° 1

## REDES DE COMPUTADORAS

Licenciatura en Ciencias de la Computación - Primer cuatrimestre de 2011  
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



- En caso de que se solicite un archivo *PHP*, el servidor deberá ejecutar el intérprete correspondiente en modo *CGI* y devolver la salida obtenida en respuesta al requerimiento efectuado.
- El programa deberá crear un mensaje de respuesta que contenga el archivo solicitado o bien el resultado de su ejecución en caso de solicitarse un archivo *PHP*.
- El directorio raíz del servidor web será por defecto el mismo donde se encuentra la imagen binaria del servidor en si mismo.
- El programa deberá considerar el caso de que el *URL* correspondiente al requerimiento no especifique archivo alguno, entonces deberá buscar por defecto y en el siguiente orden los archivos: *index.html*, *index.htm*, *index.php*, en el directorio raíz del servidor web.
- El programa deberá responder en forma adecuada cuando se presenten errores. Es decir deberá emitir los códigos de error y su descripción de acuerdo a los definidos por el protocolo *HTTP*. Para ello el alumno deberá investigar los códigos de estado del protocolo y utilizarlos en forma adecuada (*HTTP status codes*).
- El servidor web debe implementar el protocolo *HTTP 1.0*
- El servidor deberá ser capaz de devolver archivos solicitados correspondientes a los siguientes formatos gráficos: *GIF*, *PNG* y *JPG*. Deberá investigar sobre los *headers* y códigos *MIME* a especificar para la transmisión de estos archivos.
- El intérprete *PHP* (ejecutable: *php-cgi*) en modo *CGI* admite las siguientes variables de entorno definidas en la *RFC 3875*:  
`AUTH_TYPE, CONTENT_LENGTH, CONTENT_TYPE, GATEWAY_INTERFACE,  
PATH_INFO, PATH_TRANSLATED, QUERY_STRING, REMOTE_ADDR, REMOTE_HOST,  
REMOTE_IDENT, REMOTE_USER, REQUEST_METHOD, SCRIPT_NAME, SERVER_NAME,  
SERVER_PORT, SERVER_PROTOCOL y SERVER_SOFTWARE.`  
Cualquier otra no presente en la *RFC* es una extensión particular de la versión del lenguaje *PHP* que utilice.

### Consideraciones de implementación:

El programa deberá ser implementado utilizando exclusivamente el mecanismo de creación de hijos de los sistemas operativos tipo Unix, es decir la llamada al sistema *fork()*.

Las conexiones se deberán implementar bajo protocolo TCP/IP utilizando *sockets*.

El programa deberá poder finalizar automáticamente en forma normal al recibirla la señal *SIGUSR1*. Si el programa recibe otra señal se considera que no terminó en forma correcta.

Toda vez que el programa termine normalmente su ejecución se deberá informar al proceso que lo haya invocado retornando la constante *EXIT\_SUCCESS*, mediante la función de librería *exit()*. En contraste, toda terminación anormal del programa deberá informar con un código particular de error el motivo por el cual no pudo ejecutarse, y mostrar en consola una línea describiendo el error que se produjo. Dichos códigos de error y sus descripciones deberán ser documentados adecuadamente.

Recordar que deberá tener instalado en su distribución del sistema operativo GNU/Linux el intérprete para el lenguaje PHP. Para ello deberá hacer uso del administrador de paquetes de su distribución.



# PROYECTO N° 1

## REDES DE COMPUTADORAS

Licenciatura en Ciencias de la Computación - Primer cuatrimestre de 2011  
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



### Observaciones:

- En el código, entre eficiencia y claridad, se debe optar por la claridad. Toda decisión en este sentido debe constar en el informe que acompaña al programa implementado.
- Deben efectuar la programación utilizando técnicas de programación modular (incluyendo funciones y procedimientos), estudiadas durante la carrera.
- Se tendrán en cuenta aspectos relacionados con buenas técnicas de programación, simplicidad y modulación de los algoritmos implementados.
- El código entregado deberá estar comentado e indentado adecuadamente.

### Documentación:

- Deberá confeccionar la documentación orientada al programador, comentarios en el programa, y aclaraciones sobre parámetros de compilación (si los hubiese).
- Cada procedimiento/función deberá estar comentado, se deberá especificar el prototipo de la misma, explicando cada uno de los parámetros de entrada y/o salida.

### Condiciones de entrega:

- El proyecto debe ser realizado en comisiones de **dos personas (sin excepciones)**.
- El trabajo debe ser entregado el día 16 de mayo de 2010, en el horario de la clase práctica sin posibilidad de reentrega.
- El módulo principal del programa se deberá denominar: **servidorHTTP.c**
- Deberá entregarse un folio de plástico correctamente cerrado que contenga:

📁 Un cdrom o pendrive con una carpeta llamada: `proyecto01`, la cual debe contener:

- Una carpeta `bin`, con el binario del programa (**servidorHTTP**) compilado y listo para ser ejecutado.  
Una carpeta `src`, con todos los archivos fuentes junto con cualquier otro archivo adicional que se necesite para poder compilar al código fuente que dio origen al ejecutable contenido en la carpeta `bin`.

📄 El informe impreso (**con nombre y LU de cada uno de los alumnos**)

📄 El código impreso de todos los módulos implementados contenidos en la carpeta: `src`.

### Condiciones de aprobación:

- No serán corregidos programas que generen errores durante la compilación.
- No serán corregidos los proyectos donde los archivos no respeten los nombres solicitados.
- En consecuencia: proyectos **no corregidos** serán desaprobados. 🚫

### Aprobación de proyectos:

- El proyecto debe estar aprobado para el cursado de la materia.