PRÁCTICAS DE REDES 2: RECORDATORIO

Héctor Menéndez

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

5 de febrero de 2013

Index

El código

- El código debe ser claro y madurado (sin parches).
- Deben tenerse en cuenta detalles como errores, modulación, etc.

La Compilación

- Se debe poder compilar con -Wall sin errores ni warnings.
- Si la práctica no compila, se considera no entregada.

Control de Errores

- Se deben recuperar los errores recuperables, en la medida de lo posible.
- Se debe incluir en la cabecera de la función aquellos errores que sean tratados de forma especial (en los comentarios).
- Se deben mostrar al usuario a través de stderr, no de stdout.
- Se recomienda usar la librería errno.h y la función perror para informar debidamente al usuario.
- Si alguna librería facilitada por los profesores devuelve error, no perjudicará a los alumnos.

Gestión de Recursos

- Se deben liberar todos los recursos que se reserven al finalizar la ejecución, ya sea de memoria, ficheros, sockets, mutex, hilos.
- Si algún recurso no pudiera liberarse, se indicará con una nota en la cabecera de la función donde se deba liberar el recurso.

Modulación

- Se deben realizar las prácticas de forma modular, en este caso deberá haber, al menos, un módulo (definido a través de sus .c y .h) para la parte de sockets, otro para la parte de IRC y otro para la parte de usuario. Aparte, se podrán definir módulos auxiliares.

La memoria

- Las partes de la memoria son:
  1. **Introducción**: Breve resumen de lo que es la práctica de forma genérica. (Media página)
  2. **Características del Proyecto**: Manual de ejecución y ejemplos de ejecución (pocos). (Entre una hoja y una hoja y media).
  3. **Desarrollo Técnico**: Explicación técnica del proyecto. (Entre dos y tres hojas)
  4. **Resultados**: Pruebas que se han lanzado y resultados de las mismas. (Entre dos y tres hojas)
  5. **Conclusiones**: Conclusiones a las que se han llegado. (Media hoja)
  6. **Posibles Mejoras**: Posibles mejoras futuras. (Media hoja)
  7. **Bibliografía**: Referencia de textos, si se ha usado alguno. (Lo que sea necesario)
- Entre 6 y 10 páginas en total.

Manual de uso

- Debe mostrar las posibles ejecuciones del programa con sus distintas variables de forma clara y concisa.
- Debe incluir un ejemplo de ejecución de arranque del programa.

Documentación Funcional

- Se realiza con doxygen de forma automágica.
- Hay que poner los comentarios del código de forma apropiada y poner en cada función: qué hace (de forma breve), parámetros de entrada, parámetros de salida y valores de retorno. Si es necesario añadir alguna nota o comentario de la función se debe hacer en la propia cabecera.
- Hay que documentar también cada fichero y las variables globales.

Desarrollo Técnico

- Esta parte no debe hacer referencia ni a ficheros .c ó .h, ni a funciones. En ningún momento deben aparecer partes del código ni cabeceras de funciones.
- Requiere un nivel de abstracción mayor que el código, se deben mencionar módulos o capas (como podría ser la capa de transporte, la capa de aplicación o módulo IRC, el módulo de usuario, etc), y sus funcionalidades (por ejemplo: "la capa de transporte utiliza una conexión TCP capaz de..."), sin entrar en detalles de funciones.
- Debe mencionar la gestión del paralelismo, es decir, si se están utilizando subprocesos (hilos), y con qué fin, o se está utilizando un único flujo de programa.

Resultados

- Esta parte recogerá las pruebas que se realicen sobre el código pero no mostrará, por ejemplo, ficheros de trazas. Estos ficheros se incluirán en la entrega y se referenciarán en esta sección para su análisis.
- Deben realizarse las pruebas que se consideren oportunas, pero sobre todo hay que tener en cuenta la funcionalidad mínima.

Funcionalidad

- La práctica deberá ejecutarse según el manual facilitado por el alumno (y cumpliendo los requisitos del enunciado). Si la práctica no ejecuta se considera no entregada.
- Se deben evitar violaciones de segmento y errores de acceso a memoria no reservada.

Mensajes de la Red

- Los mensajes de red se deben analizar con Wireshark para ver si están bien formados. Se penalizarán los mensajes mal formados.
- Hay que tener cuidado con los puertos de origen y destino a los que se envían los mensajes y controlarlos bien.

Funcionamiento de los hilos

- Los hilos que se generen deben ser cerrados en la medida de lo posible.
- Es muy importante respetar la sincronización de los hilos, gestionar bien semáforos o mutex y liberar todos los recursos extra de los hilos cuando se dejen de usar.