

Sistemas Operativos

FIEC, ESPOL, Dra. Cristina Abad Robalino

Proyecto Parcial

Resumen del proyecto

| | |
|--|--|
| Título: | ESPOLbox |
| Estudiantes: | 1 ó 2 por proyecto |
| Puntaje: | 25 puntos (25% de la nota del parcial) |
| Fecha de entrega: | Junio 26 de 2014, 15h30 |
| Fecha de sustentación: | Junio 26 de 2014, 15h30 |
| Lenguajes de programación permitidos: | Java, C, C++ |
| Sistemas operativos permitidos: | Linux (se permiten máquinas virtuales) |
| Interfaz gráfica: | Ninguna |

Descripción y alcance del proyecto

ESPOLbox implementará una versión básica de un servicio para colocar archivos en la nube.

Requerimientos funcionales

1. Por lo pronto, solamente es necesario soportar archivos de texto (no binarios).
2. El servidor debe colocar los archivos de los usuarios en una estructura jerárquica como la siguiente (se permiten modificaciones a esta estructura, si Uds. lo consideran pertinente):

```
/ESPOLbox/usuario1/dir1/archivo1
                        dir2/archivo2
                        dir2/archivo3
                        archivo4
                        archivo5
usuario2/dir1/archivo1
...
```

3. El cliente debe interactuar con el servidor via un programa a ser ejecutado en la línea de comando. Ejemplos de interacción del servidor con el cliente:

```
% espolboxc -cuser miUserName ← Crea un nuevo usuario llamado miUserName
% espolboxc -user miUserName -setboxdir
/home/miUserName/miCarpetaDeBox ← Define cuál será la carpeta
local (del lado del cliente) que el usuario desea colocar “en la nube”; la carpeta
debe existir (de lo contrario debe mostrarse un error al usuario). Si la carpeta
ya tiene documentos dentro de ella, entonces en este momento se los debe
```

subir a la nube (a la carpeta correspondiente, según el ítem anterior). Usar mensajes amigables al usuario para que se pueda entender lo que está sucediendo. Adicionalmente, desde este momento, deberá estar corriendo un proceso en el background que periódicamente revise si es necesario sincronizar la carpeta (por ejemplo, guardar cambios hechos a archivos locales en la nube, o bajarse de la nube los cambios a archivos o nuevos archivos que hayan sido subidos desde otro PC).

`% espolboxc -user miUserName -syncbox` ← Crea una carpeta local, en el directorio actual, llamada `miCarpetaDeBox` (usar el nombre dado por el usuario, ver paso anterior) y en ese momento se baja de la nube todos los archivos y directorios del usuario y coloca una copia dentro de esta carpeta. Adicionalmente, desde este momento, deberá estar corriendo un proceso en el background que periódicamente revise si es necesario sincronizar la carpeta (por ejemplo, guardar cambios hechos a archivos locales en la nube, o bajarse de la nube los cambios a archivos o nuevos archivos que hayan sido subidos desde otro PC).

`% espolboxc -user miUserName -autosyncbox` ← Desde este momento, deberá estar corriendo un proceso en el background que periódicamente revise si es necesario sincronizar la carpeta (por ejemplo, guardar cambios hechos a archivos locales en la nube, o bajarse de la nube los cambios a archivos o nuevos archivos que hayan sido subidos desde otro PC). Este comando será utilizado por el cliente, si por alguna razón (crash, kill, etc.) ya no está corriendo en el background el proceso de sincronización.

NOTA: Para el requerimiento del proceso de sincronización en el background, también se permite la siguiente alternativa: en lugar de que el proceso corra todo el tiempo en el background, se puede usar un proceso de corta duración, que es invocado periódicamente (ej.: cada 5 minutos) por el sistema operativo. Para esto, pueden utilizar el comando `cron` de Linux. Para un tutorial del uso de “cron”, ver: <http://code.tutsplus.com/tutorials/scheduling-tasks-with-cron-jobs--net-8800>

4. Un usuario debe poder hacer cambios/modificaciones a uno de sus archivos desde una PC, y esos cambios se deberán reflejar en su carpeta ESPOLbox en otra PC.
5. Si un usuario borra un archivo de su carpeta local, durante el proceso periódico de sincronización, ese archivo también debe ser eliminado de la nube (y consecuentemente, de cualquier otra PC que ese usuario tenga sincronizada con ESPOLbox).
6. Manejo de errores adecuado y amigable al usuario.

Entregables

1. Documento impreso, a ser entregado durante la sustentación del proyecto.
2. Código y screenshots de cómo correr el cliente y servidor; entregar vía SidWeb. Ya deben haber sido subidos a SidWeb antes de la sustentación. ¡No acepto código en pen drives, CD o vía email!

Rúbrica

1. Documento de 2 páginas que explique en la primera página el diseño propuesto y en la segunda página lo ilustre gráficamente. El gráfico debe incluir los procesos definidos, los mecanismos IPC usados, etc. (5 pts).
 - a. -1 punto por cada *cuatro* faltas ortográficas o gramaticales.
 - b. *Copia o plagio*: El código a ser entregado, debe ser de la total autoría de los miembros del grupo del proyecto. Sí se permite consultar ayudas en Internet (les recomiendo stackoverflow.com), pero si utilizan código obtenido de Internet, deben: (a) citar la fuente de la que obtuvieron el código, y (b) indicar qué cambios o mejoras realizaron al mismo. La copia o plagio de proyectos es penalizada tal y como está indicado en el documento de políticas del curso.
2. Empleo correcto de múltiples procesos para realizar los requerimientos de forma colaborativa (5 puntos).
3. Empleo correcto de mecanismos IPC entre los procesos del ítem anterior (5 puntos).
4. Que resuelva el problema según los requerimientos (5 puntos).
5. Que resuelva un requerimiento adicional (hasta 5 puntos extra), como por ejemplo:
 - a. Archivos o directorios compartidos entre múltiples usuarios.
 - b. Manejo de contraseñas para los usuarios, usando PAM (Pluggable Authentication Modules) o alguna otra librería segura.
 - c. Interfaz web sencilla, que el cliente pueda usar desde otras PCs para ver o bajarse alguno de sus archivos de la nube.
 - d. Soporte de copias “en conflicto” de archivos, si estos son modificados en dos PCs semi-simultáneamente. Es decir, asegurarse que no se pierdan los cambios hechos en una PC mientras en otra PC se estaba modificando el mismo archivo (pero en una versión anterior).
 - e. Manejo de versionamiento de archivos, rollbacks y undelete.
 - f. Manual de usuario en formato “man” (y que pueda ser invocado por línea de comando usando man:
`%man espolboxc`