Sistemas Operativos
Práctica 2

1. Procesos.

Considere el programa elaborado en la práctica 1 en la que se gestiona información de perros desde una consola. Las funciones implementadas fueron: insertar, leer, borrar y buscar registros. Estos registros son almacenados en disco. Para la práctica 2, es necesario diseñar dos programas cliente-servidor, que permitan acceder a los registros almacenados en disco.

1.1. Servidor.

El programa servidor deberá gestionar el archivo *dataDogs.dat*, que almacena los registros de los perros. A su vez, el servidor podrá recibir peticiones de clientes a través de la red, para realizar cada una de las solicitudes especificadas previamente: insertar, leer, borrar y buscar. La estructura de cada registro es la siguiente:

- Nombre. Cadena de máximo 32 caracteres.
- Edad [años]. Entero de 32 bits.
- Raza. Cadena de máximo 16 caracteres.
- Estatura [cm]. Entero de 32 bits.
- Peso [Kg]. Real de 32 bits.
- Sexo [H/M]. 1 caracter.

Al entrar en ejecución, el servidor no mostrará ningún menú ni mensaje alguno. El servidor se limitará a gestionar las operaciones con los clientes y a guardar en un archivo llamado *serverDogs.log* un *log* de las operaciones que se estén realizando. El formato *log* de cada operación es el siguiente:

[Fecha YYYYMMDDTHHMMSS] Cliente [IP] [inserción | lectura | borrado | búsqueda] [registro | cadena buscada]

1.2. Cliente.

El programa cliente deberá mostrar un menú igual al de la práctica 1:

1. **Ingresar registro**. Al ingresar pide uno a uno, los campos de un registro.
2. **Ver registro**. Al ingresar muestra el número de registros presentes y solicita el número del registro a ver. Valida que el número sea válido. A partir de este menú es posible abrir para lectura y escritura la historia clínica de la mascota, que consiste en un archivo de texto plano almacenado en disco del servidor. El archivo debe ser abierto de forma automática en un editor de texto del sistema en el cliente.
3. **Borrar registro**. Al ingresar muestra el número de registros presentes y solicita el número del registro a borrar. El registro es borrado del archivo, por lo que el archivo debe reducir su tamaño.
4. **Buscar registro**. Solicita una cadena de caracteres a buscar en los campos *nombre* de los registros. Muestra todos los registros que coincidan completamente con el nombre. No se distingue mayúsculas de minúsculas.
5. **Salir**.

Cuando se digite una opción, esta deberá ser enviada al servidor a fin de iniciar su gestión. La transferencia de datos entre procesos se hará mediante la escritura directa de la estructura. El formato de los comandos es libre y deberá especificarse en el documento.

1.3. Consideraciones.

- Agrupar la información de un registro en la estructura *dogType*.
- Hacer uso de punteros y de memoria dinámica (malloc() - free()). La cantidad de memoria usada por el proceso servidor no deberá superar 1MB, por lo que la información de los animales (registro e historia clínica) debe residir en disco.
- Por cada opción ejecutada, siempre se debe dar un mensaje de confirmación y solicitar **cualquier** tecla para continuar, antes de volver al menú principal.
- Los datos de los regisrtos ingresados siempre son almacenados en un archivo llamado *dataDogs.dat* que reside en la carpeta desde donde se ejecute el programa servidor. Este archivo debe tener las características de tamaño, búsqueda y estructura, definidas en la práctica 1.
- El número máximo de clientes que se pueden conectar al servidor es de 32.

- Entrega: Archivos fuentes **p2-dogServer.c p2-dogClient.c Makefile** y archivo LEEME. Todo dentro de una carpeta con los nombres que aparecen en el correo para cada integrante. Pej: **capedrazab-capedrazab**. Incluya más archivos fuente si lo desea. Los programas principales y ejecutables deberán llamarse igual y como se indica. La forma de entrega se indicará días antes de la entrega.

1.4. Calificación.

Se tendrán en cuenta los siguientes aspectos para la evaluación:

- Funcionamiento de los programas. Servidor: 30 % Cliente: 30 %
- Código limpio. 20 % (modular, tabulaciones, comentarios - básicos, declaración de constantes, etc.)
- Documentos: manual de uso y especificaciones 10 %, informe de elaboración (funciones, diagrama de comunicaciones, diagrama de bloques) 10 %. Se tiene en cuenta redacción y ortografía de los documentos.