



Sistemas Operativos

Tarea 1:

Profesor: Viktor Tapia
Ayudantes De Cátedra: Juan Pablo Varas
Ayudantes De Tarea: Sofia Mañana

21 de marzo de 2022

1. Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje C. Se exigirá que los archivos se presenten de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso de sus programas junto a cualquier indicación que sea necesaria, y un archivo MAKE para poder ejecutar el programa.

2. Tarea

La Universidad Técnica Federico Santa María quiere tener un veterinario dentro del Campus San Joaquín, al que han denominado SansaPets. Necesitan de sus habilidades informáticas para desarrollar la plataforma online dónde se registraran los estados de las mascotas de los alumnos. Implemente una interfaz utilizando **Llamadas a Sistemas** dónde ordene los archivos dados y luego por terminal reciba instrucciones para mover, visualizar y editar los archivos antes dados.

2.1. Orden Archivos

Recibirá 10 archivos organizados de la siguiente manera:

```
nombre mascota
nombre dueño
tipo de animal
estado
```

Cada archivo tendrá el siguiente nombre: *nombre_mascota.txt*

Dónde tipo de animal puede ser: Perro, Gato, Hurón, Pájaro y Lagarto. Y estado puede ser: Internado, De Alta y Con Medicamentos. En primer lugar, se deben crear carpetas con los nombres de Tipo de Animal y ordenar los archivos correspondientes a cada tipo dentro de las carpetas. Se deben utilizar **llamadas a sistemas**, cualquier tarea que no cuenten con esto, será evaluada con la nota mínima.

2.2. Interfaz

Se debe implementar una interfaz dentro de la terminal que tenga las siguientes funciones:

1. Ver mascotas por Estado.
2. Ver mascotas por Tipo.
3. Editar Estado.
4. Editar Tipo.

Para las opciones 1 y 2 se debe poder ingresar por consola la opción y se debe desplegar los nombres de las mascotas que cumplan con la condición dada. Para las opciones 3 y 4, se debe poder ingresar el nombre de la mascota que se va a editar, el nuevo Estado o Tipo se debe ingresar por consola.

2.3. Consideraciones

- Si se edita el Tipo de una mascota, esta debe ser cambiada de carpeta.
- Se debe tener una opción para terminar el programa.
- Se darán 10 archivos para que ustedes prueben, pero a la hora de revisar se cambiarán el total de archivos y el contenido de estos, por lo que su programa debería seguir funcionando de la misma manera.
- En la interfaz, poner instrucciones **claras** de como se deben utilizar.

3. Presentación Aleatoria

Para cada tarea, se seleccionarán grupos al azar para presentar su tarea frente a ayudantes y eventualmente profesor, recibiendo una ponderación del 75 % y 25 % entre tarea y presentación respectivamente. Si su grupo presentó en una tarea, no volverá a salir nuevamente. Se comunicará días antes que grupos presentarán. Si el grupo se conforma por alumnos de paralelos distintos, nosotros comunicaremos quien debe presentar.

4. README

Debe contener como mínimo:

- Nombre, rol y paralelo de los integrantes.
- Especificación de los nombres de los archivos.
- Instrucciones generales de compilación y uso.

5. Consideraciones Generales

- Se deberá trabajar de a pares. Se deberá entregar en Aula a mas tardar el día 9 de Abril de 2022 a las 23:55 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en el lenguaje C. Se asume que usted sabe programar en este lenguaje, ha tenido vivencias con el, o que aprende con rapidez.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.
- Las tareas serán ejecutadas en **Linux**, cualquier tarea que no se pueda ejecutar en dicho sistema operativo, partirá de nota máxima 60.
- Los archivos deberán ser comprimidos y enviados juntos en un archivo .tar.gz en el formato **TAREA1_ROL1_ROL2**.
- Las preguntas deben ser hechas por Aula.
- Si no se entrega README o MAKE, o si su programa no funciona, la **nota es 0** hasta la corrección.
- Se **descontarán** 50 puntos por:
 - Mala implementación del Makefile.
 - No respetar el formato de entrega.