

Laboratorio 1: Sistemas Operativos

Profesor: Viktor Tapia
Ayudante de cátedra: Juan Pablo Varas
Ayudante de Tarea: Javiera Cárdenas

12 de Marzo 2023

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 y ejecución de sus programas junto a cualquier indicación que sea necesaria, y un archivo MAKE para poder ejecutar el programa.

2 Tarea

Se solicita desarrollar un programa que sea capaz de ordenar distintos juegos de Steam, representados por un archivo cada uno, con respecto a diferentes parámetros que le serán entregados. Luego de ordenarlos, su programa también deberá ser capaz de mostrar datos con respecto al mismo orden. Estas dos partes de la tarea deberán ser entregadas en **archivos de código distintos**, cada uno con nombre acorde al punto que se resolverá, como se explicará a continuación.

2.1 Orden Archivos

Recibirá 20 archivos organizados de la siguiente manera:

Cantidad de jugadores actual

Mayor cantidad de jugadores

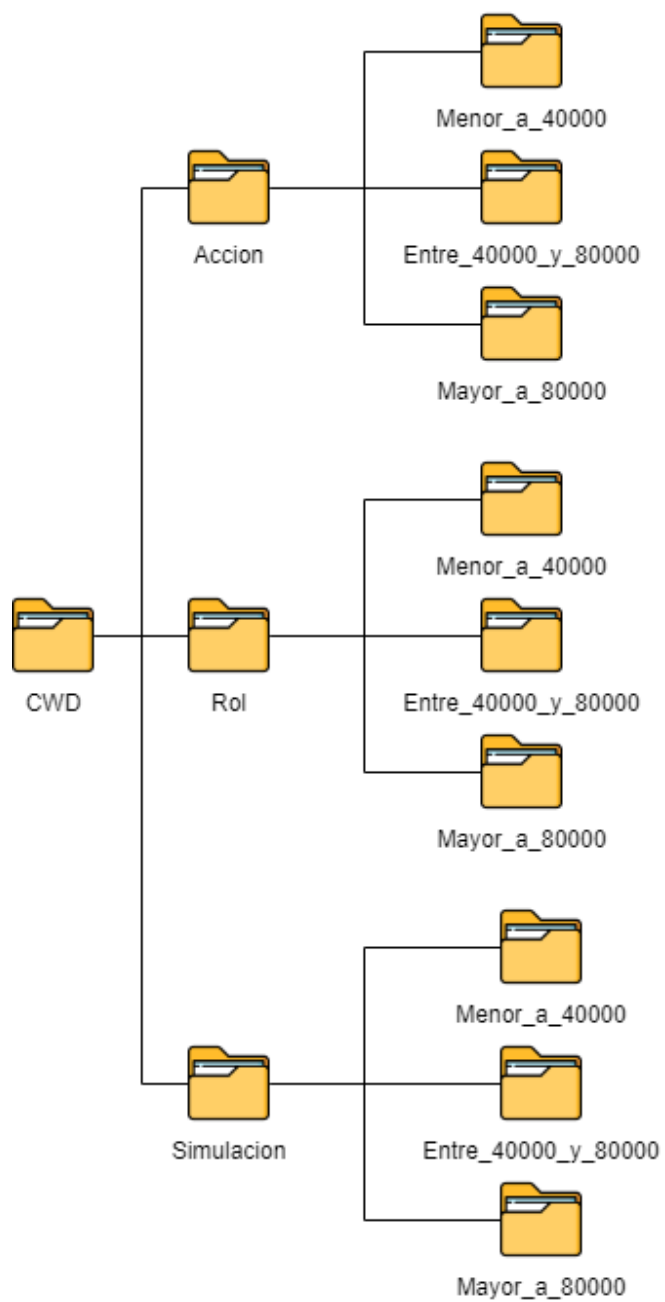
Género

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

Donde la cantidad de jugadores corresponde al número de jugadores conectados jugando ese juego en específico, la mayor cantidad de jugadores corresponde a la mayor cantidad de jugadores que tuvo ese juego en el día y el género al cual pertenece el juego.

En primer lugar, se deben crear directorios que correspondan a cada género que tengan los juegos y ordenar los archivos correspondientes a cada juego dentro de los directorios. Posteriormente dentro de cada uno de esos directorios se deben ordenar los juegos en relación a un input recibido a través de la consola, que puede ser "cantidad actual" para indicar que los juegos deben ser ordenados a partir de la cantidad de jugadores actual o "mayor cantidad" para ordenarlos por la mayor cantidad de jugadores en los rangos de "*Menor_a_40000*", "*Entre_40000_y_80000*" y "*Mayor_a_80000*" para ambos casos.

Se adjunta la siguiente imagen para mayor claridad con un ejemplo sobre la organización directorios:



2.2 Visualización por consola

En esta segunda parte del problema, el programa correspondiente deberá buscar los archivos de texto (ya ordenados previamente por la sección anterior), y entregar estadísticas válidas con respecto a la categoría indicada con el siguiente formato:

- Cantidad de juegos por género.
 - Cantidad de juegos por rango de jugadores, que sean parte de ese género.
- Cantidad de juegos por rango de jugadores.

Esto debe ser mostrado en consola como se da en los ejemplos de la siguientes imagenes:

```
Cantidad de juegos de acción :
  Juegos con una cantidad de jugadores actuales menor a 40000: 5
  Juegos con una cantidad de jugadores actuales entre 40000 y 80000: 3
  Juegos con una cantidad de jugadores actuales mayor a 80000: 2
(. . .)

Del total de juegos:
  Existen 6 con una cantidad de jugadores actuales menor a 40000
  Existen 5 con una cantidad de jugadores actuales entre 40000 y 80000
  Existen 3 con una cantidad de jugadores actuales mayor a 80000
```

Caso 1: Archivos ordenados por la cantidad de jugadores actual

```
Cantidad de juegos de acción :
  Juegos con una mayor cantidad de jugadores menor a 40000: 4
  Juegos con una mayor cantidad de jugadores entre 40000 y 80000: 6
  Juegos con una mayor cantidad de jugadores mayor a 80000: 3
(. . .)

Del total de juegos:
  Existen 5 con una mayor cantidad de jugadores menor a 40000
  Existen 7 con una mayor cantidad de jugadores entre 40000 y 80000
  Existen 4 con una mayor cantidad de jugadores mayor a 80000
```

Caso 2: Archivos ordenados por la mayor cantidad de jugadores

2.3 Consideraciones

- Se darán 20 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.
- Se pueden tener hasta 4 géneros distintos en los archivos.
- Para facilitar la realización del programa, cada juego solo tendrá solo un género.

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 80% y 20% entre tarea y presentación respectivamente. Si su grupo presentó en una tarea, no volverá a salir nuevamente.

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 XX de Marzo de 2023 a las 23:59 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. De esta forma los demás grupos pueden beneficiarse en base a la pregunta, **se responderán consultas hasta 48 hrs. antes de la fecha y hora de entrega**.
- 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.