Universidad Simón Bolívar
Departamento de Computación y
Tecnología de la Información
CI-3825: Sistemas de Operación I

Proyecto 2 Enero – Marzo 2023

Los amigos de Ash Ketchum

1 Introducción

Luego de más de 25 años, la serie de televisión *Pokémon* ha llegado a su fin. La saga de Ash Ketchum (centro imagen anterior) ha terminado.

A lo largo de la serie, Ash conoce muchos interesantes personajes, la mayoría de los cuales deja con la intención de verlos de nuevo. Pero, ¿a cuántos logró revisitar? Por suerte, *Pokémon* tiene fans que documentan obsesivamente a todos y cada uno de los personajes. Por mala suerte, no son muy buenos contando.

La información de cada personaje se resume en un documento HTML. Entre más grande sea el documento, más información se sabe del personaje (generalmente indicando que Ash ha interactuado más con él). Los personajes se clasifican por

- Región geográfica
- Especie (pokémon ó humano)
- Número de apariciones (clasificado de forma discreta en:
    - Principal– indicando que Ash interactúa con ellos en todos los episodios;
    - Recurrente – indicando que Ash, en efecto, los vuelve a visitar;
    - Líder de Gimnasio – indicando que Ash tiene una relación más profesional con ellos; y
    - Apareciendo una sola vez – indicando que Ash, en efecto, no los vuelve a visitar)

En ese orden.

Este sistema de clasificación jerárquico hace difícil contabilizar los personajes, por lo que se le pide que haga un programa que pueda realizar las cuentas de diferentes consultas sobre ellos.

2 Requerimientos del programa

Debe crear un programa que pueda ejecutarse en la carpeta raiz del directorio de archivos HTML, desde la cónsola con el comando:

```
$ ./fameChecker    [-r <region>] [-s <species>] [-t <type>]
                   [-c|--nocount] [-l|--list] [--size] [name]
```

donde

- -r indica que debe limitarse la búsqueda a una región. El nombre de la región debe llevar la misma grafía que el directorio que la contiene.
- -s indica la especie (en este caso, la especie debe ser “pokemon” para indicar que se debe contar solamente los archivos HTML en los directorios de pokémones que correspondan con los criterios de búsqueda o “trainer” para indicar que se debe contar solamente a los humanos
- -t indica el tipo de apariciones: “main” para los personajes principales, “recurring” para los personajes recurrentes, “gym_leader” para los líderes de gimnasio, y “one_time” para los personajes que aparecen solo una vez
- -c ó --nocount indica si debe aparecer el número de archivos encontrados. (Por defecto, se muestra. Si el flag está presente, no se debe mostrar.)
- -l ó --list indica si se deben dar los nombres de los archivos encontrados. (Por defecto, no se muestran.)
- --size indica si se debe mostrar el tamaño en kilobytes (1024 bytes) de los archivos. (Por defecto, no se muestra.) Si está incluido el flag -l, se muestra junto a cada archivo encontrado, su tamaño
- [name] permite restringir la búsqueda a archivos que comiencen con el nombre dado

2.1 Llamadas de ejemplo:

Suponga que queremos saber cuántos pokémones pertenecientes a Ash existen que sólo aparecen una vez en todas las regiones. En este caso, podemos hacer la búsqueda:

```
$ ./fameChecker -s pokemon -t one_time “Ash's”
```

Si queremos saber cuáles son los líderes de gimnasio de Kanto, pero no nos interesa el número:

```
$ ./fameChecker -t gym_leader -r kanto --nocount -l
```

Si necesitamos tener una idea de cuánta información puede haber sobre un personaje que aparece solo una vez:

```
$ ./fameChecker -t one_time --list --size
```

Si deseamos averiguar cuánta información total hay sobre los personajes principales:

```
$ ./fameChecker -t main --size
```

Obsérvese que estas consultas son complicadas de hacer con `grep` ó `ls`.

3 Requerimientos del informe

Debe incluir un informe que contenga:

- Introducción
- Estructura del programa y decisiones de diseño
- Cualquier cambio en la implementación o resultado sorprendente
- Conclusiones (incluyendo su respuesta a la pregunta motivante) y recomendaciones

4 Requerimientos de la entrega

Debe entregar su código en un archivo tar.gz ó .tgz que contenga únicamente:

- Archivos .c
- Archivos .h
- Makefile
- Informe

No se debe incluir el directorio HTML ni archivos .o

El proyecto debe ser subido al Moodle de la materia en la sección marcada como “ Proyecto 2”. Sólo deberá efectuar una entrega por grupo.

5 Evaluación

El proyecto tiene una ponderación de 15 puntos. Se asignarán

- 4 puntos por código:
  - 1 punto por acceso correcto a los i-nodos de archivos
  - 1 punto por acceso correcto a los i-nodos de directorios
  - 1 punto por su manejo de flags
  - 1 punto por su método de contabilización
- 7 puntos por ejecución (un punto por manejar cada uno de los *flags* correctamente, incluyendo la impresión del número correcto en cada caso)
- 4 puntos por el informe (un punto por cada sección)

El programa debe correr sin errores y seguir todas las convenciones de programación de C en Unix