

Curso Laboratorio II 2021

Ejercicio de clase Semana 7

1. Competencia de grupos – programa *anagramas-hash*

- a. El programa (cuya función main ya está hecha) lista los anagramas de una lista de palabras que se pasarán como argumentos, por ejemplo:

```
> anagramas-hash marejada volar aparte  
marejada: majadear majadera marejada  
volar: valor volar  
aparte: aparte patear patera tapera
```

- b. La lista de todas las palabras se extrae de un diccionario. El diccionario completo en español que usa este programa está en `/usr/share/dict/spanish`.
- c. Los anagramas son aquellas palabras que tienen las mismas letras, pero en diferente orden, por ejemplo: “rato” y “otra”.
- d. Se deduce del punto anterior, que dos palabras son anagramas cuando al ordenar sus letras, el resultado coincide. Por ejemplo, al ordenar las letras de “rato” y “otra”, en ambos casos se llega al resultado “aort”.
- e. Para entender y poder hacer este ejercicio, es un requisito que lean la sección 6.6 del libro de K&R, en donde se explican las básicas de la inserción en una estructura (tabla) de *hash* y se proporciona un ejercicio de ejemplo, parecido a esta propuesta. Hay infinidad de artículos que explican el funcionamiento y la conveniencia de este tipo de tablas, por ejemplo: https://en.wikipedia.org/wiki/Hash_table
- f. El programa entonces hace lo siguiente (el programa principal ya está provisto):
- Lee todo el diccionario y para cada palabra genera una “clave” ordenando sus letras. Luego aplica una función de hash para cada clave y con ese resultado inserta cada palabra en la tabla de hash.
 - Para cada uno de los argumentos suministrados en la línea de comandos, genera la clave y luego busca la palabra. Si la clave no existe, la palabra no existirá en el diccionario. También puede pasar que la clave sí exista, pero no exista palabra con dicha clave (por ejemplo: “rtoa” sería una clave válida, pero una palabra inexistente).
- g. Se proporciona el directorio `/var/local/labii2021/anagramas-hash/` que tendrá varios archivos que se usarán en este proyecto.

- h. Por ejemplo, las definiciones de estructuras para usar y las funciones a desarrollar están en el archivo `/var/local/labii2021/anagramas-hash/anagramas-hash.h`
- i. El programa principal (la función `main`) se encuentra en `/var/local/labii2021/anagramas-hash/anagramas-hash.c`
- j. La consigna para los grupos es desarrollar las demás funciones declaradas en `anagramas-hash.h`
- k. Debe hacerse un `Makefile` para compilar (`make anagramas-hash`)
- l. Verán también otros programas fuente, por ejemplo, `fgetword-normalizada.c` tiene un par de funciones que permiten leer palabras de un archivo abierto que devuelve en *word* la siguiente palabra y le saca tildes y ñes, convirtiéndolas en vocales y enes. La segunda además pasa todo a minúsculas y elimina cualquier carácter no alfabético. Esto se usa para considerar una vocal con tilde para los anagramas. Por ejemplo, *ábaco* es anagrama de *caoba*. No deben preocuparse por los detalles de la programación de `fgetword-normalizada.c`.
 - i. `extern char * fgetword_normalizada (FILE *fp, char *word, int maxword);`
 - ii. `extern char * fgetword_lower_alpha_normal(FILE *fp, char *word, int maxword)`
- m. También se dejan los programas fuente: `hash.c`, `wrappers.h`, `wrappers.c`. Los “wrappers” protegen al programador de errores comunes en funciones de la biblioteca estándar (por ejemplo, para *malloc* el wrapper se llama *malloc_or_exit*) porque ejecutan la función y controlan el valor de retorno. En caso de error, el programa finaliza y no sigue adelante.
- n. Grupos (a sortear):
 - i. Grupo1: a completar
 - ii. Grupo2: a completar
 - iii. Grupo3: a completar
 - iv. Grupo4: a completar
 - v. Grupo5: a completar
 - vi. Grupo6: a completar
 - vii. Grupo7: a completar
 - viii. Grupo8: a completar
 - ix. Grupo9: a completar