

75.41 Algoritmos y Programación II Curso 4

TP1 - El Cuervo de Tres Ojos

Memoria dinámica, punteros a funciones y otras cositas

23 de agosto de 2019



Fecha de presentación: 20/08/2019

Fecha de vencimiento: 03/09/2019

1. Introducción

El **Cuervo de Tres Ojos** es un personaje misterioso ligado a los Antiguos Dioses de más allá del Muro. De acuerdo a los Hijos del Bosque es el último de los verdevidentes, nombre atribuido a los hombres sabios que poseen habilidades mágicas que incluyen la vista verde y la capacidad de ser cambiapieles (pueden entrar en la mente de algunos animales y controlarlos a voluntad).

Según las historias, una persona con vista verde sueña como el común de la gente, pero cuando ocurre un "sueño verde", éste está lleno de significados simbólicos, imágenes y metáforas de eventos pasados, presentes y futuros que son interpretados como visiones.

El Cuervo presenta un cuerpo esquelético y ropas putrefactas que hacen que, a simple vista, pueda ser confundido con un cadáver. Sin embargo, se trata de un hombre tan antiguo que las raíces de un arciano han crecido sobre, bajo y a través de él. Lo poco de piel que se puede observar es blanca, salvo por una marca roja sangre que atraviesa su cuello y llega a su mejilla. Su cabello blanco es fino y ligero, y suficientemente largo para barrer el piso.

Cuando habla sus palabras son acompañadas por un leve rumor de hojas y ramas, y un ligero balanceo de su cabeza. Su voz es seca y sus labios se mueven lentamente, como si hubieran olvidado cómo formar palabras. En la narrativa de Juego de Tronos, tras despertar del coma inducido por su caída de la torre, **Brandon Stark** percibe la aparición del Cuervo múltiples veces en sus sueños hasta que, en cierta ocasión, este le advierte que nunca volverá a caminar, pero que puede enseñarle a volar.

Mucho tiempo después, con la ayuda de Jojen Reed, Meera y Hodor, Bran llega a una caverna al norte del Muro, donde es guiado por una hija del bosque hacia donde se encuentra el Cuervo de Tres Ojos. Allí, comienza a instruir al joven Stark en la verdevidencia y el cambio de pieles. Llegado el momento, el Cuervo decide que es hora de dar el próximo paso y le da a Bran un cuenco de pasta de arciano para despertar sus poderes de verdevidente y que, de esta manera, se convierta eventualmente en su sucesor.

2. Objetivo

El presente trabajo tiene como finalidad que el alumno repase algunos de los conocimientos adquiridos en Algoritmos y Programación I, así como también que comience a familiarizarse con las herramientas a utilizar a lo largo de la materia, tales como el manejo de memoria dinámica y la utilización de punteros a funciones.

3. Enunciado

El Cuervo de Tres Ojos quiere transmitirle a su aprendiz, Bran, la mayor cantidad de visiones que pueda. Sin embargo, como es demasiada información para que el joven adquiera, decide comenzar inculcándole un conjunto de visiones del pasado y del futuro que comparten determinadas características.

Se pide crear una biblioteca cuervos (.h y .c) que permita cargar y manipular la transmisión de dichas visiones. Particularmente se espera que se implementen, dentro de la biblioteca, las siguientes funciones:

```

1 /*
2  * Recibe el nombre del cuervo a invocar.
3  * Creará la estructura cuervo_tres_ojos, reservando la memoria necesaria.
4  * Devolverá la estructura con los topes de las visiones inicializados en 0.
5  * En caso de no poder invocar al cuervo, devolverá NULL.
6 */
7 cuervo_tres_ojos_t* invocar_cuervo(char* nombre);
8
9 /*
10 * Recibe el nombre del aprendiz a invocar.
11 * Creará la estructura cuervo_aprendiz, reservando la memoria necesaria.
12 * Devolverá la estructura con el tope de las visiones inicializado en 0.
13 * En caso de no poder invocar al aprendiz, devolverá NULL.
14 */
15 cuervo_aprendiz_t* invocar_aprendiz(char* nombre);
16
17 /*
18 * Agrega una vision al vector de visiones correspondiente.
19 * Reservando la memoria necesaria para ella.
20 * Devolverá 0 si pudo agregarla o -1 si no pudo.
21 */
22 int agregar_vision(cuervo_tres_ojos_t* cuervo, vision_t vision);
23
24 /*
25 * Transmitirá al aprendiz las visiones del pasado y del futuro del cuervo que sean
26 * transmisibles de acuerdo a la función recibida como parámetro.
27 * En caso de que la visión a transmitir ya sea conocida por el aprendiz (repetida),
28 * no se agrega a las visiones del mismo.

```

```

29 * Devolverá 0 si se transmitieron con eso, -1 en caso contrario
30 */
31 int transmitir_visiones(cuervo_tres_ojos_t* cuervo, cuervo_aprendiz_t* aprendiz, bool (*
32     es_transmisible)(vision_t));
33 /*
34 * Mostrará por pantalla las visiones adquiridas por el aprendiz.
35 * Una visión se listará sólo si la función recibida como parámetro lo permite.
36 */
37 void listar_visiones_transmitidas(cuervo_aprendiz_t aprendiz, bool (*es_listable)(vision_t));
38 /*
39 * Destruirá la estructura del cuervo, liberando la memoria reservada para él y sus visiones.
40 * Devolverá si pudo liberar la memoria o -1 si no pudo.
41 */
42 int destruir_cuervo(cuervo_tres_ojos_t* cuervo);
43 /*
44 * Destruirá la estructura del aprendiz, liberando la memoria reservada para él y sus visiones.
45 * Devolverá si pudo liberar la memoria o -1 si no pudo.
46 */
47 int destruir_aprendiz(cuervo_aprendiz_t* aprendiz);

```

A su vez, se cuenta con los siguientes registros a utilizar en el presente trabajo:

```

1 typedef struct vision {
2     char protagonista[MAX_NOMBRE];
3     char casa_protagonista[MAX_NOMBRE];
4     char descripcion[MAX_VISION];
5     char epoca; // P - pasado, F - futuro
6 } vision_t;
7
8 typedef struct cuervo_tres_ojos {
9     char nombre[MAX_NOMBRE];
10    vision_t* visiones_del_pasado;
11    vision_t* visiones_del_futuro;
12    int cantidad_visiones_pasado;
13    int cantidad_visiones_futuro;
14 } cuervo_tres_ojos_t;
15
16 typedef struct cuervo_aprendiz {
17     char nombre[MAX_NOMBRE];
18     vision_t* visiones_adquiridas;
19     int cantidad_visiones;
20 } cuervo_aprendiz_t;

```

Las visiones se encontraran en el archivo visiones.txt. El mismo contiene 4 datos por linea, separados entre si por un punto y coma. Dichos datos se corresponden a los encontrados en el struct vision anteriormente mostrado, siendo ellos: Protagonista de la vision, la casa de dicho protagonista, una descripcion de la vision, un caracter que indica si la epoca de la vision es pasado o futuro (P o F, respectivamente)

Siguiendo este formato, son ejemplos validos:

```

1 Jon Snow;Stark;Se une a la Guardia de la Noche;P
2 Jon Snow;Stark;Viaja más allá del Muro;P
3 Daenerys Targaryen;Targaryen;Se convierte en la madre de dragones;P
4 Harry Potter;Gryffindor;No supera que lo cambien por un Hobbit en Algo1 y se infiltra en Algo2;F

```

4. Resultado esperado

Se espera que el trabajo creado sea compilado sin errores con la siguiente línea:

```
1 gcc *.c -Wall -Werror -Wconversion -std=c99 -o cuervo_de_tres_ojos
```

Luego, que sea ejecutado, cargue e imprima por pantalla las visiones transmitidas.

Para verificar el correcto funcionamiento, debe utilizarse un programa principal con el siguiente flujo:

```

1 /*
2 * Funciones que determinan si es transmitible o listable:
3 * - Como mínimo es necesario crear 3.
4 * - Ejemplos:
5 * -- Una función que devuelva true si es del pasado o false si no.
6 * -- Una función que devuelva true si es del futuro o false si no.
7 * -- Una función que devuelva true si es Jon Snow o false si no.
8 * -- Una función que devuelva true si la visión tiene más de 50 letras.
9 * -- Etc. sean creativos!

```

```

10 */
11
12 int main(){
13     /* Abrir el archivo */
14     /* Crear cuervo y aprendiz */
15     /* Cargar visiones al cuervo */
16     /* Cerrar el archivo */
17     /* Transmitir visiones del cuervo al aprendiz */
18     /* Puede ser invocada varias veces con diferentes funciones, */
19     /* esto hará que el cuervo aprendiz tenga distintas visiones de distintos rubros */
20     /* Mostrar las visiones adquiridas por el aprendiz */
21     /* Puede ser invocada varias veces con diferentes funciones, */
22     /* esto hará que el listado varíe dependiendo de lo que se le */
23     /* transmitió y de lo que se le pide listar */
24     return 0;
25 }

```

Y, ser corrido con **valgrind**:

```
1 valgrind --leak-check=full --track-origins=yes --show-reachable=yes ./cuervo_de_tres_ojos
```

Por ejemplo, si partimos del siguiente archivo **visiones.txt**

```

1 Jon Snow;Stark;Le dicen que no sabe nada;P
2 Daenerys Targaryen;Targaryen;Se convierte en la madre de dragones;P
3 Daenerys Targaryen;Targaryen;Viaja a Poniente;F
4 Jon Snow;Stark;¡Era por abajo Pacacio!;F
5 Tyrion Lannister;Lannister;Mata a su propio padre;F
6 Jon Snow;Stark;Se une a la Guardia de la Noche;P

```

Y transmitiendo las visiones de Jon se obtendrá el siguiente resultado.

```

1 ==4056== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
2 ==4056== Command: ./cuervo_de_tres_ojos
3 ==4056==
4 Visión sobre Jon Snow: Le dicen que no sabe nada
5 Visión sobre Jon Snow: ¡Era por abajo Palacio!
6 Visión sobre Jon Snow: Se une a la Guardia de la Noche
7 ==4056==
8 ==4056== HEAP SUMMARY:
9 ==4056==     in use at exit: 0 bytes in 0 blocks
10 ==4056==   total heap usage: NN allocs, NN frees, MM,MMM bytes allocated
11 ==4056==
12 ==4056== All heap blocks were freed -- no leaks are possible
13 ==4056==

```

5. Referencias

- https://gameofthrones.fandom.com/wiki/Three-Eyed_Raven
- https://hieloyfuego.fandom.com/wiki/Cuervo_de_tres_ojos
- <https://hieloyfuego.fandom.com/wiki/Verdevidente>
- https://hieloyfuego.fandom.com/wiki/Vista_verde