CS3026 Assesment 1 Memory

Stefan Nikolov

November 1, 2019

1. CGS D3 – D1
   1.1 Requirements – gcc compiler
   1.2 Instructions
   - Open terminal and navigate to the directory which contains the source files or ***right-click*** while in the folder and press ***Open Terminal Here.***
   - Run the following command: ***make***
   - Run the following command: ***./shell***

   1.3 Explanation

   We're creating an array which represents our memory and a linked list which acts as a table for our segments. Then we are printing them.

   1.4 Output

   The output should look something like this

```
[ 0] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 10] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 20] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 30] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 40] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 50] 00 00 00 00 00 00 00 00 00 00 | ..........
```

   Etc. …

```
[1010] 00 00 00 00 00 00 00 00 00 00 | ..........
[1020] 00 00 00 00
```

   Segment 0

   Allocated = FALSE

   Start = a hexadecimal address

   Size = 1024

2. CGS C3 – C1
   2.1 Instructions
   - Open terminal and navigate to the directory which contains the source files or ***right-click*** while in the folder and press ***Open Terminal Here.***
   - Run the following command: ***make***
   - Run the following command: ***./shell***

   2.2 Explanation

   We're allocating 50 bytes of our memory using mymalloc(), a custom version of the system malloc() to three different pointers. Then we are printing the memory array and segment table to show how they've changed

   2.3 Output

   The output should look something like this

Please see next page

```
[ 0] 00 00 00 00 00 00 00 00 00 00 | this test1
[ 10] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 20] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 30] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 40] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 50] 00 00 00 00 00 00 00 00 00 00 | this test2
Etc..
[100] 00 00 00 00 00 00 00 00 00 00 | this test3
Etc..

[1010] 00 00 00 00 00 00 00 00 00 00 | ..........
[1020] 00 00 00 00
```

Segment 0

Allocated = True

Start = starting hex address of mymemory array

Size = 50

Segment 1

Allocated = True

Start = start of previous segment + size

Size = 50

Segment 2

Allocated = True

Start = start of previous segment + size

Size = 50

Segment 3

Allocated = FALSE

Start = a hexadecimal address

Size = 874

3. CGS B3 – B1

3.1 Instructions

- Open terminal and navigate to the directory which contains the source files or ***right-click*** while in the folder and press ***Open Terminal Here.***
- Run the following command: ***make***
- Run the following command: ***./shell***

3.2 Explanation

After we have created a segment and allocated a certain amount of memory for it we are going to remove it. In this case we are removing pointer2 with content -> this test2

3.3 Output

The output should look something like this

Please see next page

```
[ 0] 00 00 00 00 00 00 00 00 00 00 | this test1
[ 10] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 20] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 30] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 40] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 50] 00 00 00 00 00 00 00 00 00 00 | ..........
Etc..
[100] 00 00 00 00 00 00 00 00 00 00 | this test3
Etc..
[1010] 00 00 00 00 00 00 00 00 00 00 | ..........
[1020] 00 00 00 00
```

Segment 0

Allocated = True

Start = starting hex address of mymemory array

Size = 50

Segment 1

Allocated = False

Start = start of previous segment + size

Size = 50

Segment 2

Allocated = True

Start = start of previous segment + size

Size = 50

Segment 3

Allocated = FALSE

Start = a hexadecimal address

Size = 874

4. CGS A5 – A1

4.1 Instructions

- Open terminal and navigate to the directory which contains the source files or ***right-click*** while in the folder and press ***Open Terminal Here.***
- Run the following command: ***make***
- Run the following command: ***./shell***

4.2 Explanation

Once we have removed pointer2 we have external fragmentation which means that there's space between pointer1 and pointer3 which is unused, and our memory must be defragged. After defragging we are adding a new pointer to our memory so that we can see that it is working as intended.

4.3 Output

The output should look something like this

Please see next page

```
[ 0] 00 00 00 00 00 00 00 00 00 00 | this test1
[ 10] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 20] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 30] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 40] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 50] 00 00 00 00 00 00 00 00 00 00 | this test3
Etc..
[100] 00 00 00 00 00 00 00 00 00 00 | ..........
Etc..
[1010] 00 00 00 00 00 00 00 00 00 00 | ..........
[1020] 00 00 00 00
```

Segment 0

- **Allocated = True**
- **Start = starting hex address of mymemory array**
- **Size = 50**

Segment 1

- **Allocated = True**
- **Start = start of previous segment + size**
- **Size = 50**

Segment 2

- **Allocated = FALSE**
- **Start = a hexadecimal address**
- **Size = 924**

```
[ 0] 00 00 00 00 00 00 00 00 00 00 | this test1
[ 10] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 20] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 30] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 40] 00 00 00 00 00 00 00 00 00 00 | ..........
[ 50] 00 00 00 00 00 00 00 00 00 00 | this test3
Etc..
[100] 00 00 00 00 00 00 00 00 00 00 | this test4
Etc..
[1010] 00 00 00 00 00 00 00 00 00 00 | ..........
[1020] 00 00 00 00
```

Segment 0

- **Allocated = True**
- **Start = starting hex address of mymemory array**
- **Size = 50**

Segment 1

- **Allocated = True**
- **Start = start of previous segment + size**
- **Size = 50**

Segment 2

- **Allocated = True**
- **Start = start of previous segment + size**
- **Size = 50**

Segment 3

- **Allocated = FALSE**
- **Start = start of previous segment + size**
- **Size = 874**

4.4 Notes on mydefrag():

- When trying to implement the code for the case where we've removed pointer1, a segmentation fault occurs right after the statement – printf("mydefrag> start\n"); and I could not find a solution to that problem.