Using Memory Leak Detector

This page discusses the steps on on how to use the memory leak detector code.

All the supportive codes are available in this directory: **$CANVAS_COURSE_REFERENCE**

$$CANVAS_COURSE_REFERENCE$$

CANVAS_COURSE_REFERENCE$/files/folder/Dynamic%20Memory%20Allocation/Memory%20leak%20detector (https://webcourses.ucf.edu/courses/1390330/files/folder/Dynamic%20Memory%20Allocation/Memory%20leak%20detector)

Follow the following steps to use the memory leak detector code

Step 1: In order to detect memory leak, you need the following supporting files:

a. leak_detector_c.c

b. leak_detector_c.h

Step 2: Add the following lines in your code:

a. add the following line immediately after including all the header files

```
#include "leak_detector_c.h" //immediately after including all the head
er files
```

b. add the following line to the beginning of the main function of your code:

```
atexit(report_mem_leak); //add this line to your code
```

Let's say the file name of your code is main.c

Step 3: Now compile the leak_detector_c.c and main.c together in the same command

Step 4: Execute your code and you will see a txt file **"leak_info.txt"** where you will find all the memory leaks and line numbers of your code. If there is no memory leak, it will be an empty txt file.

Depending on your IDE and working environment you might get error for not compiling the leak_detector_c.c file.

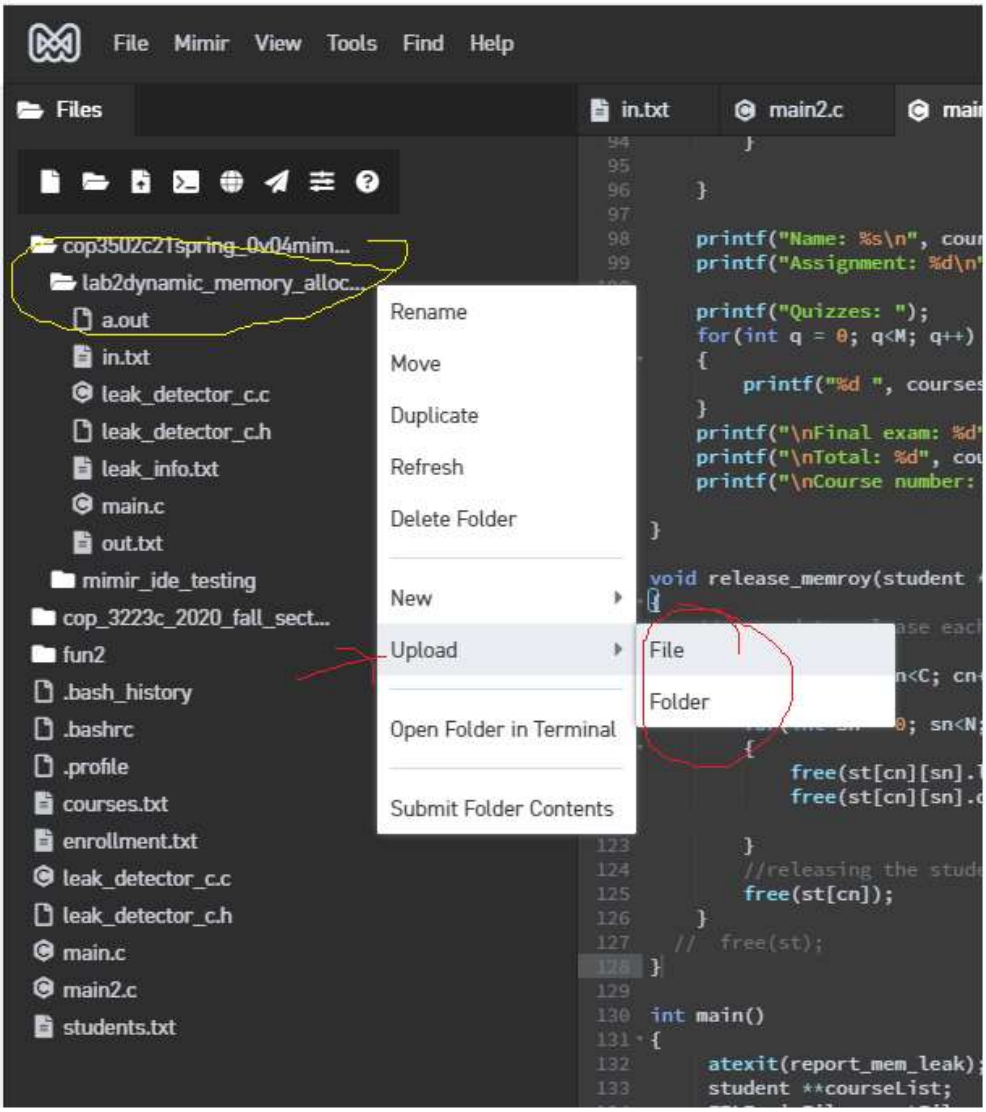
You need to compile both ***leak_detector_c.c*** and ***your .c*** files to get the result.

In repl.it:

Add all the files while coding in the repl.it. It will compile and run smoothly as repl.it will compile both of the files when you press the run button.

If you use Mimir IDE or other command based system:

1. Upload the "leak_detector_c.h", "leak_detector_c.c" and yourcode.c (if any) to the directory. See the example image bellow (right-click the folder and upload files to that folder)



2. run the following commands:

$ gcc leak_detector_c.c yourfilename.c

3. If the code compiles then run:

$ ./a.out

After running a.out, your code should produce output. It will also create file "leak_info.txt". If you open the file, you will see the memory leaks, if any.

If you are using linux, write the command $ls , it will show you all the files in your directory. If you want to see the content of leak_info.txt file, wrote the command $cat leak_info.txt, you will be able to see the content in this file.

For Other IDE:

If you see an IDE such as codeblocks, you can also run your code in code blocks or other IDEs. But, in that case you have create a project and add all the files to the project. In that case codeblocks/other IDE will be able to compile all the c files in the project.

In all cases, whatever compiler/IDE you use, make sure that your code works in mimir.