Valgrind

(last updated 2/9/18)

Valgrind is a program to that detects memory issues resulted from the execution of your program. It is available on Linux systems (from the terminal). (Thanks to Jake Nissley for recommending this tool.)

This program is already on omega. Once you produced your executable program, instead of running it with:

```
./myprog
```

run:

```
valgrind --leak-check=yes ./myprog
```

will show a report at the end similar to:

```
==1210==
==1210== HEAP SUMMARY:
==1210==     in use at exit: 0 bytes in 0 blocks
==1210==   total heap usage: 142 allocs, 142 frees, 2,272 bytes allocated
==1210==
==1210== All heap blocks were freed -- no leaks are possible
==1210==
==1210== For counts of detected and suppressed errors, rerun with: -v
==1210== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```

Here is a sample Valgrind report of **memory leaks**. (Your program should NOT have this):

```
==27047== 400 (16 direct, 384 indirect) bytes in 1 blocks are definitely lost in
loss record 3 of 3
```

Her is a sample Valgrind report of **invalid memory access**. (Your program should NOT have this):

```
==9814==  <font color = red>Invalid write of size 1</font>
==9814==    at 0x804841E: main (example2.c:6)
==9814==  Address 0x1BA3607A is 0 bytes after a block of size 10 alloc'd
```

To show the line numbers for the errors reported by Valgrind:

1. compile with the -g flag:
   ```
   gcc -g -o hw6 hw6.c
   ```

2. run with `=full` instead of `= yes`.
   a. Here is a run with user input:
   ```
   valgrind --leak-check=full ./myprogr
   ```
   b. Here is a run with file redirection:
   ```
   valgrind --leak-check=full ./myprogr < data.txt
   ```

*** Here is another interesting example. What causes the `fileread_test.c` program to have a memory leak ? Get the files: fileread_test.c, fileread_data.txt?

Valgrind information:

- Quick Start Guide
- Read more about using Valgrind and understanding the errors it reports on this page from Cprogramming.com.
- Here is the link to the valgrid tutorial that I based the above example on. There are more resources on the web.
- "Intermediate C Programming" by Lu, Yung-Hsiang, isbn: 9781498711630. Book recommended by Dr. Brezeale that explains Valgrind (the book covers other topics such as how to make makefiles). NOTE that this book is NOT required and we will not make any references to it. (Thanks to Scott Laue for the reference)