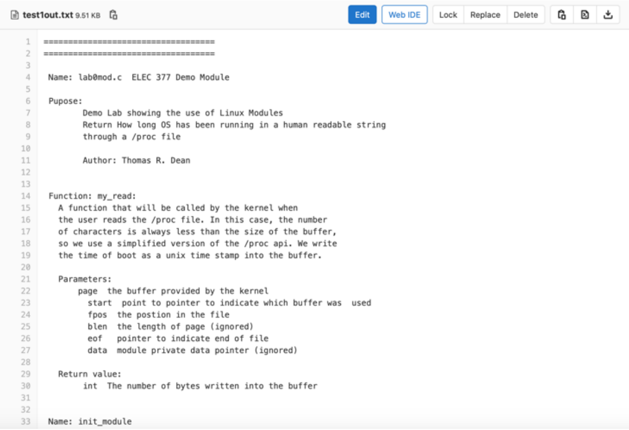
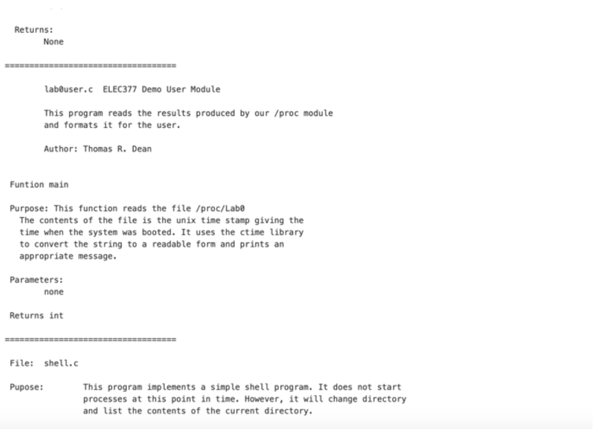
Test Description

Script 1



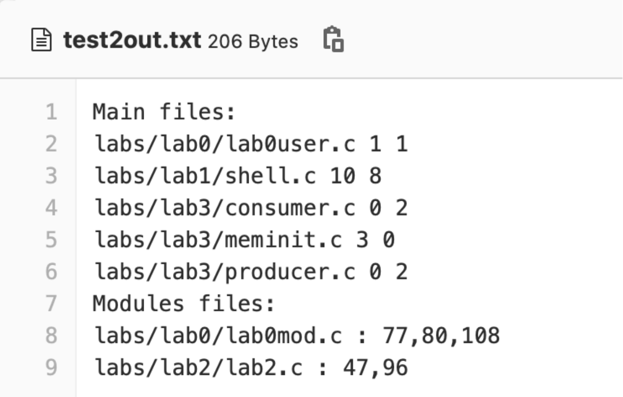
```
===================================
===================================

 Name: lab0mod.c  ELEC 377 Demo Module

 Pupose:
        Demo Lab showing the use of Linux Modules
        Return How long OS has been running in a human readable string
        through a /proc file

        Author: Thomas R. Dean


 Function: my_read:
   A function that will be called by the kernel when
   the user reads the /proc file. In this case, the number
   of characters is always less than the size of the buffer,
   so we use a simplified version of the /proc api. We write
   the time of boot as a unix time stamp into the buffer.

   Parameters:
       page  the buffer provided by the kernel
         start  point to pointer to indicate which buffer was  used
         fpos  the postion in the file
         blen  the length of page (ignored)
         eof   pointer to indicate end of file
         data  module private data pointer (ignored)

   Return value:
        int  The number of bytes written into the buffer


 Name: init_module
```



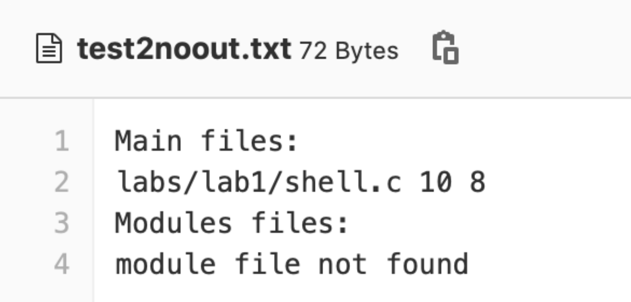
```
  Returns:
        None

====================================

        lab0user.c  ELEC377 Demo User Module

        This program reads the results produced by our /proc module
        and formats it for the user.

        Author: Thomas R. Dean


 Funtion main

 Purpose: This function reads the file /proc/Lab0
   The contents of the file is the unix time stamp giving the
   time when the system was booted. It uses the ctime library
   to convert the string to a readable form and prints an
   appropriate message.

 Parameters:
        none

 Returns int

====================================

 File:  shell.c

 Pupose:        This program implements a simple shell program. It does not start
                processes at this point in time. However, it will change directory
                and list the contents of the current directory.
```

As seen in the outputs, the test file starts by the bars of equal signs, one is the initial bar and the other is the one followed by each extracted file. All the comments in the .c or .h files are extracted into the test file; these comments are will organized. Between each file, a bar of equal signs is also added as expected.

Script 2



```
Main files:
labs/lab0/lab0user.c 1 1
labs/lab1/shell.c 10 8
labs/lab3/consumer.c 0 2
labs/lab3/meminit.c 3 0
labs/lab3/producer.c 0 2
Modules files:
labs/lab0/lab0mod.c : 77,80,108
labs/lab2/lab2.c : 47,96
```



```
Main files:
labs/lab1/shell.c 10 8
Modules files:
module file not found
```

As seen in the pictures above, when the argument number is great than 0, the file path of both main files and module files are be printed, for the main files, the number of printf and fprintf are shown after the file path. Also, for module files, the lines number of printk is printed after the file as expected.

For the one with no output of module files, main file is printed as expected, but the module file shows the error message that no file is found.