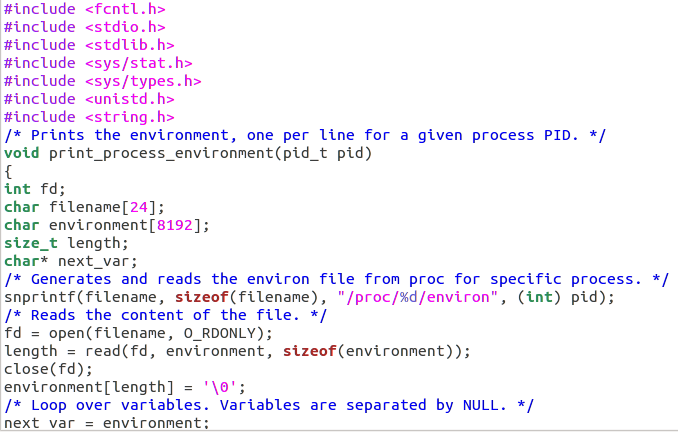
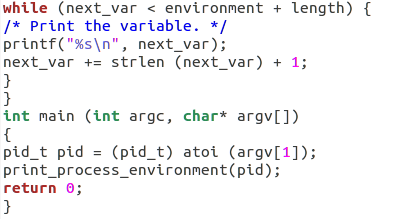
## Due: Week 11

## Grade: /30

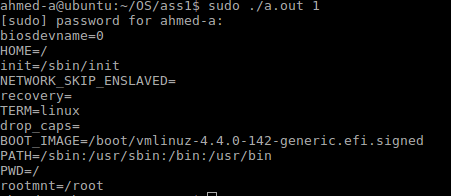
## Instructions:

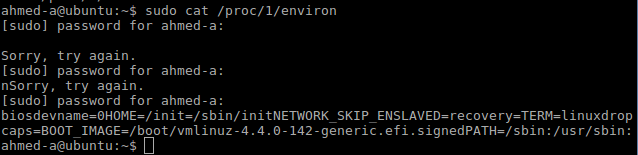
1. (10 points) The following C program will extract the environment from **proc pseudo file system** for a provided process PID.





* 1. Compile and run this program for a specific process PID.
  2. Analyze the program and results. Compare the results with /proc/pid/environ





* 1. Create a C program that:
     1. Clones a process
     2. Gets new process PID (getpid())
     3. Extracts the status from proc directory for cloned process (implement similar code used to extract process environment)
     4. Child process sends a signal to parent process (implement kill or signal( ) system call). Verify if the parent received the signal
     5. Attached code and results.

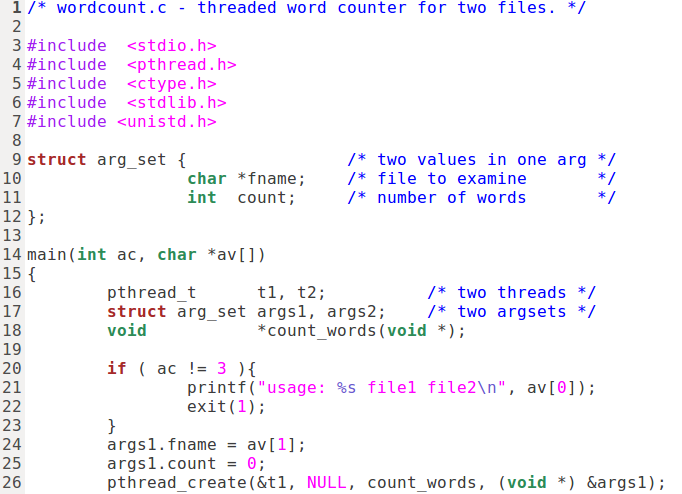
1. (10 points ) Modern operating systems are multithreaded. A thread is an instance (function) of a process. These functions (threads) run in parallel sharing resources. In this exercise we will use POSIX threads.
   1. Read Linux man pthreads and identify what is shared and what is unique by threads.

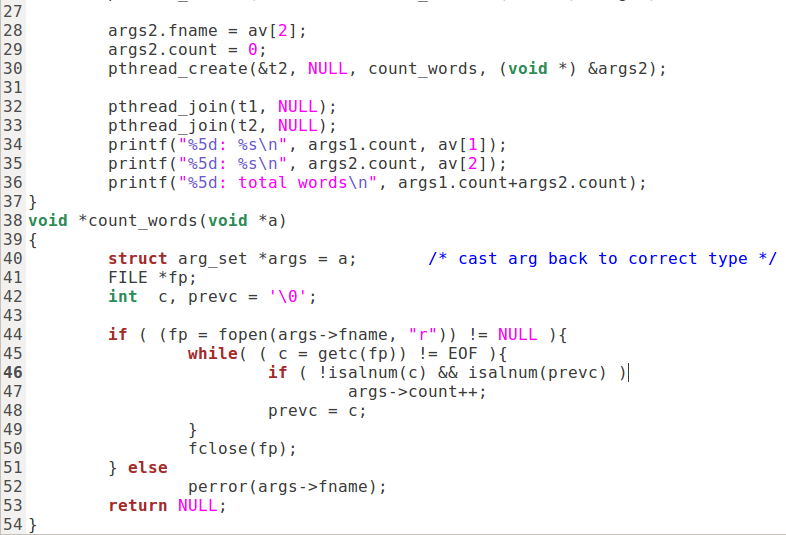
* These threads share the same global memory (data and heap segments), but each thread has its own stack (automatic variables).
  1. Read Linux manual to learn how to implement pthread\_create( ), pthread\_join( ) system calls. These system calls are in chapter (3).

**man 3 pthread\_create**

* 1. Compile and run the following program. To compile pthreads the library pthread has to be linked manually during compilation process.

gcc threadname.c -lpthread

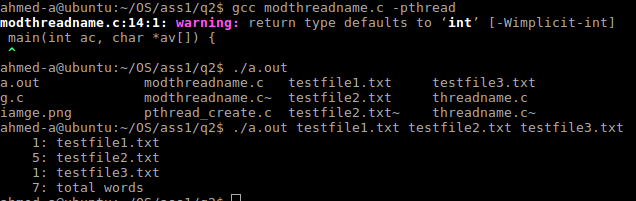




* 1. Analyze the code and results.

The code parses through to files by putting names of the files in the arguments. Creates 2 threads and prints out the amount of words in each files and total amount of words from both files..

* 1. Add a third thread to count the words of a third file.
  2. Attach modified program and results



* 1. Create a C program that :
     1. Clones a process
     2. The child process will create two POSIX threads. One thread will get input from STDIN ( getchar() ) and print the respective ASCII code of the input character and the second thread will count and print the number of input characters
  2. Attach the program and results

1. (10 points ) Explore sysdig tool.
   1. Read sysdig documentation <https://sysdig.com/opensource/inspect/>
   2. Install and explore the tool
   3. Provide in detail three features you learned about this tool.
   4. Can be this tool used in security investigation or forensics? Explain your answer.