Lab VI Linux System Programming

* **Objective**
* Capable of using system calls to perform file operation, process management and inter-process communication
* Capable of using gcc to compile source code
* Capable of using makefile to manage and compile big programms
* **Lab Environment**
* Ubuntu Desktop
* GNU GCC
* **Instructions**
  + Use this file as a template to finish your lab report.
  + Your lab report should be renamed into LabIV\_ID.rar which contains this doc file as well as the scripts you create
  + For each of the following tasks, you have to take screen shot of the terminal window to prove that you finish the task, and paste it right below the related question
  + Using the attached source file as starting points
* **Tasks**

1. Using gcc to compile the source code: fork\_CPUtime.c

Requirements:

1） Define macro variable LOOP and assign 1，10，100，1000，10000， 100000 and 1000000 it.

2） For each LOOP, run the resultant executable file and compare the outputs

3） Describe the mechanisms of CPU resource allocation of Linux system

1. Use gcc to compile source file pipeexample.c. Run the executable to help you understand the mechanism of inter-process communication using pipe. Based on the output, describe how file descriptors are assigned.
2. Use gcc to compile source file kernelManage.c and specify its output as readwrite.exe. Run the executable file and explain the mechanism how Kernel manages the core resources
3. Write a makefile that allows to selectively compile the source codes: fork\_CUPtime.c, pipeexample.c and kernalMamage.c.

Requirements:

1) by default, make compiles all the source codes;

2) command make pipe lets you compile pipeexample.c;