1. Operating System Concept Chapter 2 Exercises: 2.9, 2.10, 2.12. 2.17 (40 points)

2. Compile and run the following code and capture the running results. (20 points)

   https://github.com/zryfish/ostep/blob/master/p1.c
   https://github.com/zryfish/ostep/blob/master/p2.c
   https://github.com/zryfish/ostep/blob/master/p3.c
   https://github.com/zryfish/ostep/blob/master/p4.c

3. Expand the ptrace sample code used in the class to display the ***pathname*** parameters of the ***open*** system call.(40 points)

   Hints: a) how the parameters are passed in the open system call? b) use ***PTRACE_PEEKDATA*** to obtain the data from the process that is currently being traced. You cannot just dereference the pointer to obtain the data. Otherwise, a segment fault will occur (why?)

   You need to submit the source code in one file (say ptrace.c or other names you like, but in **one** file). The source code should be compiled on the ubuntu 18.04 (64 bits) with the gcc compiler.

   Reference:
   http://blog.rchapman.org/posts/Linux_System_Call_Table_for_x86_64/

   The ptrace sample code demonstrated in the class is shown in the following.

```
#include <sys/ptrace.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/user.h>

#include <syscall.h>

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>


#if __WORDSIZE == 64
#define REG(reg) reg.orig_rax
#else
#define REG(reg) reg.orig_eax
#endif

int main(int argc, char* argv[]) {
  pid_t child;

  if (argc == 1) {
    exit(0);
  }
  char* chargs[argc];
  int i = 0;

  while (i < argc - 1) {
    chargs[i] = argv[i+1];
    i++;
  }
  chargs[i] = NULL;

  child = fork();
  if(child == 0) {
    ptrace(PTRACE_TRACEME, 0, NULL, NULL);
    execvp(chargs[0], chargs);
  } else {
    int status;

    while(waitpid(child, &status, 0) && ! WIFEXITED(status)) {
      struct user_regs_struct regs;
      ptrace(PTRACE_GETREGS, child, NULL, &regs);
      fprintf(stderr, "system call %zd from pid %d\n", (size_t) (REG(regs)), child);
      ptrace(PTRACE_SYSCALL, child, NULL, NULL);
    }
  }
  return 0;
}
```