DUE & DEMO 4-9-2020
Turn in your zip file to TA

READ: Chapter 7.7.5: fork-exec in Unix/Linux
      Chapter 7.7.6: implementation of fork
      Chapter 7.7.7:Implementation of exec

1. BASE CODE: your LAB5pre step 3.

2. Download files from samples/LAB6/USER into a USER directory
   mku u1 ===> generates /bin/u1 on sdimage

```
/*********** u1.c contents *****************/
include "ucode.c"

int main(int argc, char *argv[])
{
  int i, pid, ppid, r;
  char line[64]; char uc;

  for (i=0; i < argc; i++){
    printf("argv[%d] = %s\n", i, argv[i]);
  }

  while(1){
    pid  = getpid();
    ppid = getppid();
    printf("This is process %d in Umode at %x parent=%d\n",
           pid, getPA(), ppid);

    umenu();
    printf("input a command : ");
    ugets(line);
    uprintf("\n");

    if (strcmp(line, "ps")==0)
      ups();
    if (strcmp(line, "switch")==0)
      uswitch();
    if (strcmp(line, "exit")==0)
      uexit();

    if (strcmp(line, "wait")==0)
      uwait();
    if (strcmp(line, "fork")==0)
      ufork();
    if (strcmp(line, "exec")==0)
      uexec();
  }
}


//-------------- ucode.c has these syscalls ----------------------
int ufork()
{
  int pid;

  pid = syscall(9,0,0,0); // syscall 9 is for fork()
  if (pid>0){
    uprintf("parent %d forked a child %d\n", getpid(), pid);
  }
  if (pid==0){
    uprintf("child %d return from fork(), pid=%d\n", getpid(), pid);
  }
  if (pid < 0)
    uprintf("%d fork failed\n", getpid());
}

int uexec()
{
  int r;
  char line[64];

  uprintf("enter a command string : ");
  ugets(line);
  uprintf("line=%s\n", line);
  r = syscall(10,line,0,0); // syscall 11 is for exec(line)
  if (r<0)
     uprintf("%d exec failed\n", getpid());
}
//--------------------------------------------------------------
```

So, you must add syscall numbers 9 and 10 to your svc.c file

                        REQUIREMENTS

PART 1: 50 points

1. Implement int fork() in kernel.
2. Implement int exec(char *cmdLine) in kernel.
          where cmdLine = input line from user for exec, e.g.
                exec
                enter a cmdLine: u2 one two three <== cmdline to exec().

PART2: 50 points

In Chapter 7, the sample code of fork/exec (which is YOUR PART 1) assume each
Umode image size is 1 MB at
                           7MB + pid*1MB.
Modify it to: Each Umode image size is 2MB at the physical addresses
          First 1MB at: 7MB + pid*1MB
          Second 1MB at: 16MB + pid*1MB

Sample solution: samples/LAB6/t.bin