CS460 LAB Assignment 5
DUE & DEMO 3-26-2020

1. BASE CODE: your LAB5pre step 3.

2. Download files from samples/LAB5/USER into a USER directory
   mku u1 ===> generates /bin/u1 on sdimage

```
/*********** u1.c contents *****************/
int main()
{
  int pid, ppid;
  char line[64];

  pid = getpid();

  while(1){
    printf("This is proc %d in Umode at %x parent=%d\n",
                        pid, getPA(), getppid());
    umenu();
    printf("input a command : ");
    ugets(line);

    if (strcmp(line, "getpid")==0)
       ugetpid();
    if (strcmp(line, "getppid")==0)
       ugetppid();
    if (strcmp(line, "ps")==0)
       ups();
    if (strcmp(line, "chname")==0)
       uchname();
    if (strcmp(line, "switch")==0)
       uswitch();
// The above commands are supported in your LAB5pre



// ADD these commands and syscalls to kernel
    if (strcmp(line, "sleep")==0)
       usleep();
    if (strcmp(line, "wakeup")==0)
       uwakeup();
    if (strcmp(line, "kfork")==0)
       ukfork();
    if (strcmp(line, "exit")==0)
       uexit();
    if (strcmp(line, "wait")==0)
      uwait();
  }
}

int umenu()
{
  uprintf("---------------------------------------------\n");
  uprintf("ps chname switch sleep wakeup kfork exit wait\n");
  uprintf("---------------------------------------------\n");
}



// ADD these syscalls to kernel
int usleep()
{
  int pid = getpid();
  if (pid==1){
    printf("P1 does not sleep in Umode\n");
    return -1;
  }
  printf("proc %d go to sleep in kernel\n", pid);
  return syscall(5, pid, 0, 0);
}

int uwakeup()
{
  int pid;
  printf("enter a pid to wakeup: ");
  pid = geti();
  printf("pid=%d\n", pid);
  return syscall(6,pid,0,0);
}

int ukfork()
{
  return syscall(7, "u1", 0, 0);
}

int uexit()
{
  int value;
  printf("enter an exit value : ");
  value = geti();
  syscall(8, value, 0, 0);
}

int uwait()
{
  int pid, status;
  pid = syscall(9, &status, 0, 0);
  printf("pid = %d ", pid);
  if (pid > 0)
    printf("status = %x",status);
  printf("\n");
}
```

==========

Therefore, you MUST modify svc.c in kernel
to support these syscalls.

==========

In t.c: P0 code:

```
   // after initialize kernel

   kfork("u1");

   kprintf("P0 switch to P1\n");   ==> run P1 in Umode on /bin/u1 file

   while(1){
     if (readyQueue)
       tswitch();
   }
```

sample solution: samples/LAB5/solution, sdimage