Student 1 - kmitt006
Student 2 - zwen012

Lab 1 Writeup

- usys.S
  - Added a SYSCALL(waitpid) as that's a new system call
- syscall.h
  - Added a SYS_waitpid with a call number of 22 as it's a new system call
- syscall.c
  - Created a sys_waitpid function call that is called with there's a system call to waitpid
  - Set up the sys_waitpid function to be called with SYS_waitpid is called
- sysproc.c
  - Changed sys_exit so it calls our updated exit function, in addition sys_exit now has some argument cleaning(checks for invalid arguments)
  - Changed sys_wait so it calls our updated wait function, in addition sys_wait now has some argument cleaning(checks for invalid arguments)
  - Created the definition of sys_waitpid which calls our new waitpid function, and has argument cleaning(checks for invalid arguments)
- proc.c
  - Adjusted the exit function so that when status is passed in, the current process's status is set to the passed in parameter
  - Adjusted the wait function so that when status is passed in, it checks to see if the pointer passed in exists, if so, it assigns the process's status(which is the child process) to the argument passed in.
  - Created the waitpid function, which does the same thing as wait, except now, we have to check to see if the process that is passed in, is the process we're looking for, this means that now, we need to check to see if the process we find in the table is the process we're looking for, if so we should just stop. In addition, we have a semi-implemented WNOHANG working.
- proc.h
  - Had to add a status variable to the proc object so I could do stuff with status in exit, wait, and waitpid
- defs.h
  - Modified the declaration for the updated exit and wait function
  - Added a declaration for the waitpid function
- user.h
  - Modified the declaration for the updated exit and wait function
  - Added a declaration for the waitpid function
- test_lab1.c
  - This is just the TA test code
- Additional Notes:
  - We modified every single call of exit() and wait() to exit(0) and wait(0) when necessary