Zachery Creech

Dr. Beck

COSC361

2 May 2021

Project 3

Modifications to the following files:

Makefile:

- Addition of “umalloc.o” to compilation command for _forktest as recommended on Piazza to stop compilation error with malloc() and free()

proc.c:

- Addition of two new system calls clone() and join() for working with threads
    - clone() works almost identically to fork(), but rather than creating a child with its own address space create a thread with a shared address space
        - See proc.c near the bottom for implementation and comments
    - join() works almost identically to wait() but it checks for threads instead of children
        - See proc.c at the bottom for implementation and comments
- Changes to growproc() to account for updating threads’ address spaces along with parent process’s
    - See proc.c for comments
- Changes to exit() to account for threads exiting differently than children
    - See proc.c for comments
- Changes to wait() to account for waiting only for children, not for threads. join() is for waiting for threads
    - See proc.c for comments

proc.h:

- Addition of member variables “num_threads” and “stack” to proc data structure to keep track of a process’s number of threads and stack
    - void *stack;
    - int num_threads;

sysproc.c:

- Addition of two functions sys_clone(void) and sys_join(void) to do trap security checking
    - See sysproc.c at the bottom

syscall.c:

- Addition of external sys_ function declarations for clone() and join()
    - extern int sys_clone(void);
      extern int sys_join(void);
- Addition of sys_ function tags in static syscalls[]
    - [SYS_mprotect] sys_clone,
      [SYS_munprotect] sys_join,

defs.h:

- Addition of function signatures for clone() and join()
    - int clone(void(*)(void *, void *), void *, void *, void *);
      int join(void **);

user.h:

- Addition of struct definition for type lock_t

    -
      ```
      typedef struct __lock_t
      {
        uint flag;
      } lock_t;
      ```
- Addition of function signatures for clone() and join()
    - int clone(void(*)(void *, void *), void *, void *, void *)
      int join(void **);
- Addition of function signatures for thread_create() and thread_join()
    - int thread_create(void(*)(void *, void *), void *, void *);

        - int thread_join();

syscall.h:

- Define trap numbers for SYS_clone and SYS_join
    - #define SYS_clone 22
      #define SYS_join 23

usys.S:

- Addition of clone() and join() to SYSCALL script list
    - SYSCALL(clone)
      SYSCALL(join)

ulib.c:

- Addition of implementations for user locking functions lock_init(), lock_acquire(), and lock_release()
    - See ulib.c near the bottom for comments
- Addition of implementations for user thread functions create_thread() and join_thread() that internally call system calls clone() and join() respectively
    - See ulib.c at the bottom for commments

I found this project to be quite a bit more difficult than both Projects 1 and 2. Copying fork() and wait() was not too confusing since the only real differences were comparing address spaces, but tracing memory back to growproc() to account for malloc(), as well as dealing with the changes to exit() and wait() were extremely confusing to me. I am honestly still not sure I even did it right. In fact, I am pretty convinced I didn't.

In the same vein, I really could not figure out a good way to test that everything was working. The only test I could come up with basically just hung on execution, and I have no idea if that was due to the test being wrong or due to the project implementation itself. I chose not to submit it because it didn't work anyway. I think that I got the general idea of the changes that needed to be made, but I'm doubtful that I actually did them all correctly.