

SSE3044 Introduction to Operating Systems  
Prof. Jinkyu Jeong

## Project 3. Synchronization

---

2018.05.16 (Wed.)

TAs

이규선([gyusun.lee@csl.skku.edu](mailto:gyusun.lee@csl.skku.edu)) /

안민우([minwoo.ahn@csl.skku.edu](mailto:minwoo.ahn@csl.skku.edu))

# Project Plan

---

- Total 4 projects

- 1) Process management

- System call

- Priority scheduler

- 2) Virtual memory

- 3) Synchronization

- Thread (5/16 ~ 5/29 11:59pm)

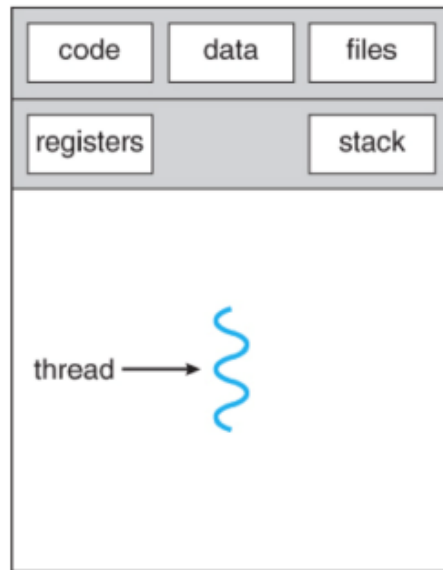
- Mutex (TBD)

- 4) File system

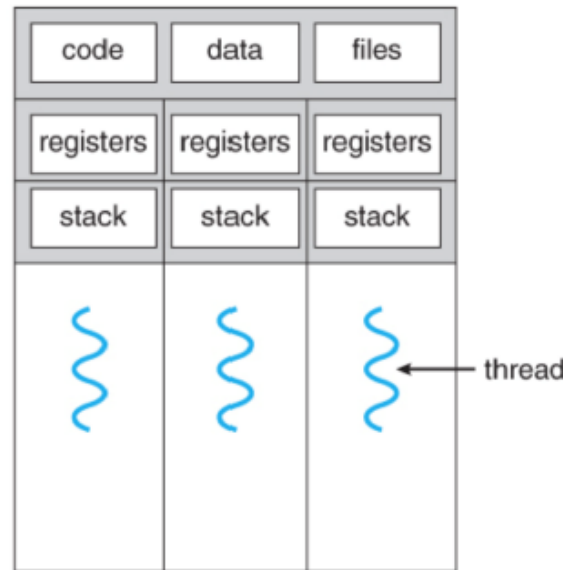
# Supporting Threads on Xv6

---

- The original xv6 process is single-thread
- Multi-thread environment
  - Each thread has its own stack
  - Every threads shares code, data and other resources such as open files

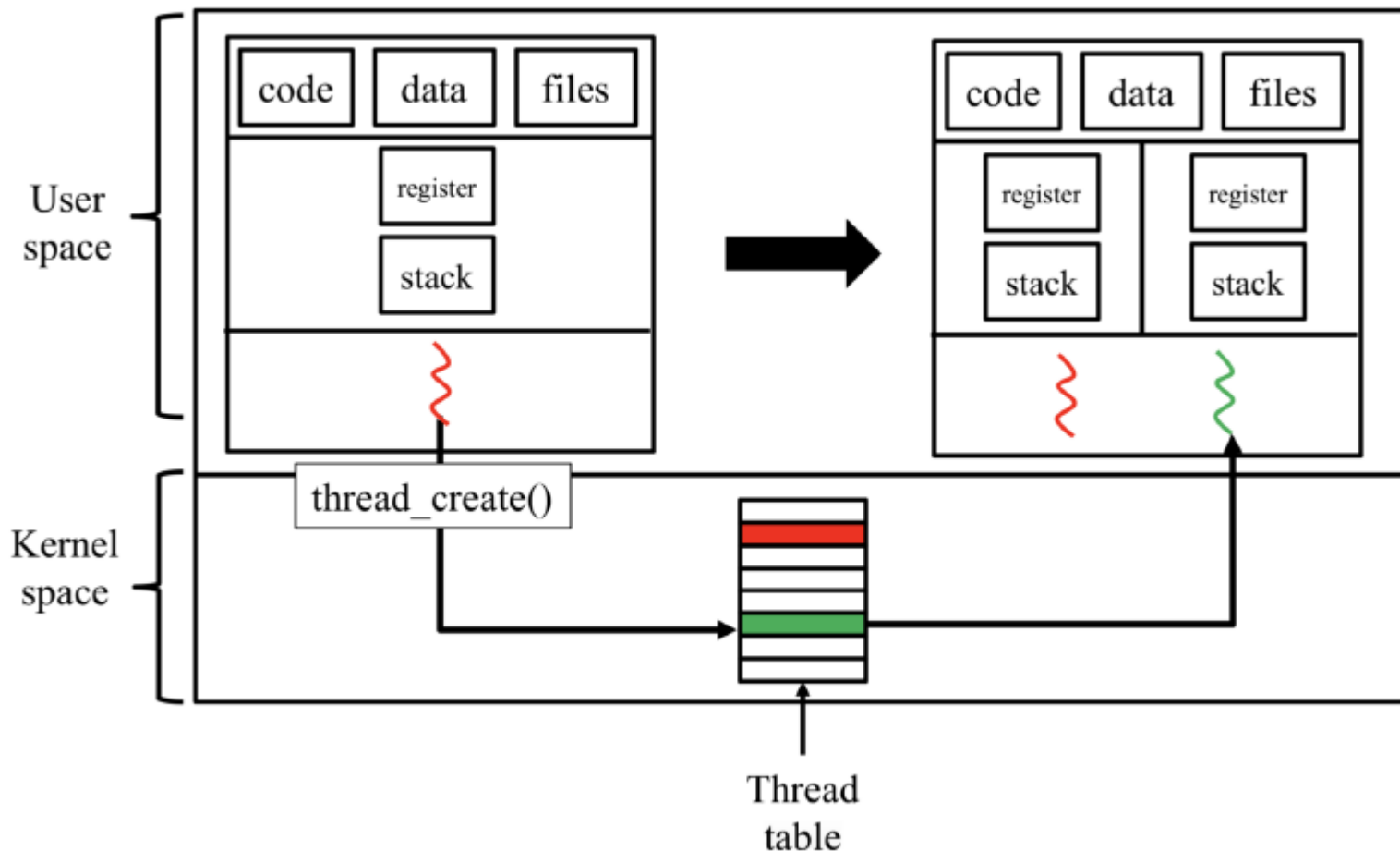


single-threaded process



multithreaded process

# Supporting Threads on Xv6



# Project 3-1. Thread

---

- Make 3 API with thread supporting
  - `int thread_create(void *(*function)(void *), void *arg, void *stack)`
  - `void thread_exit(void *retval)`
  - `int thread_join(int tid, void **retval)`
- Implement method for getting thread ID
  - `int gettid(void)`

# thread\_create()

---

- Synopsis

- Create a new thread at calling process
- `int thread_create(void *(*function)(void *), void *arg, void *stack)`

- Return value

- Return the thread ID(tid) of the new thread
- If err, return -1

# thread\_create()

---

- The new thread starts execution in invoking **function**.
- **Arg** is passed as the argument of **function**.
- **Stack** is the pointer to call stack of new thread.
- All threads in a process have same pid & priority.
- Initial thread in a process is a parent of others & has tid '1'.
- A process can have maximum 8 threads.

# thread\_create()

---

- Synopsis
  - Create a new thread at calling process
- Return value
  - Return the thread ID(tid) of the new thread
  - If err, return -1



# thread\_exit ()

---

- Synopsis
  - Terminate calling thread
  - void thread\_exit(void \*retval)

# thread\_exit ()

---

- Each thread save **retval** at thread\_exit().
- Thread state transfers to ZOMBIE.
- Thread resources are retrieved at thread\_join.
- Exiting thread may wake up parent thread.

# thread\_join ()

---

- Synopsis

- Join with terminated thread
- `thread_join(int tid, void **retval)`

- Return value

- If success, return 0
- If there's no thread with input tid, return -1

# thread\_join ()

---

- Wait thread specified with **tid** to terminate.
  - Caller may sleep until corresponding thread terminated.
  - If thread has already terminated, return immediately.
- Copy the exit status of the target thread into the location pointed to by **retval**.
- The call stack of the terminated thread should be freed by the calling thread.
  - Resources should be retrieved at this point.

# gettid()

---

- Returns caller's thread ID.
- In multi-thread process, all threads have the same PID.
- Each thread has a unique TID within a process.

# Tips

---

- If the main thread terminates or any thread calls `exit()`, whole process is terminated. In this case, all threads should be terminated as well. Also, address space should be freed and open files should be closed.
- Open files are shared among threads. If thread A opens a file, the file can be also accessed by another thread B(in the same process) using same file descriptor. Files opened by thread A need not be closed automatically when thread A terminates.

# Tips

---

- When a thread calls `thread_exit()`, the thread remains in ZOMBIE state until another thread calls `thread_join()`.
- There is no parent-child relationship among threads. Any thread can invoke `thread_join()` for another thread.
- All threads within a process should return the same process ID. Thread IDs are guaranteed to be unique only within a process.

# Submission

---

- Compress your xv6 folder as YourStudentID-3-1.tar.gz
  - `tar cvf 2016710580-3-1.tar.gz xv6-SSE3044`
- Send your tar.gz file to [gyusun.lee@csl.skku.edu](mailto:gyusun.lee@csl.skku.edu)
  - Please command `$make clean`, before submission
  - Please send mail with uniformized title
    - [SSE3044]YourStudentID-3-1
- **PLEASE DO NOT COPY**
  - **YOU WILL GET F GRADE IF YOU COPIED**
- Due date: 5/29(Tue.), 23:59:59 PM
  - -25% per day for delayed submission



# Grade policy

---

- If you failed to pass “oral test”, you will get 0
- Describe how you have implemented to pass each test case (If not, we consider it as failed to pass test cases)

# Questions

---

- If you have questions, please email to TA
- You can also visit Semiconductor Building #400509
  - Please email TA before visiting