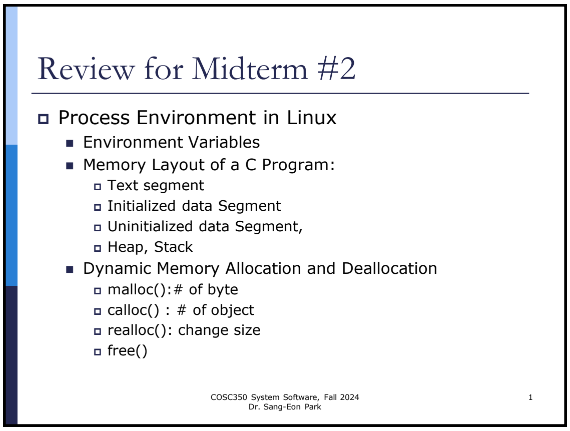
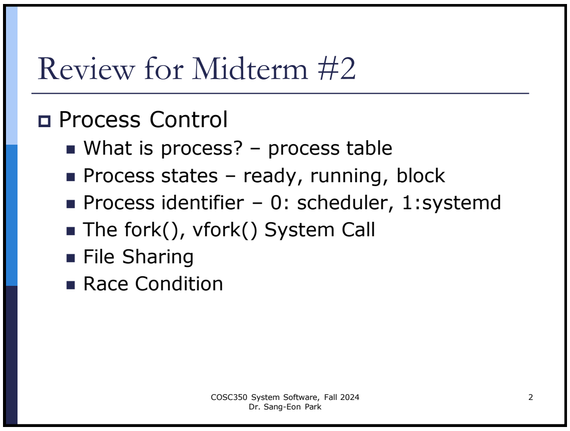


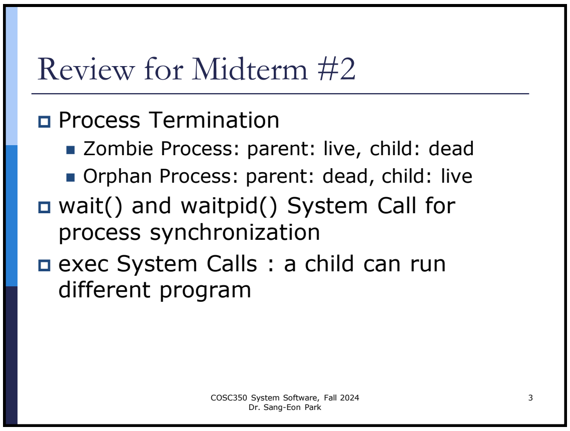
Review for Midterm #2

- Process Environment in Linux
  - Environment Variables
  - Memory Layout of a C Program:
    - Text segment
    - Initialized data Segment
    - Uninitialized data Segment,
    - Heap, Stack
  - Dynamic Memory Allocation and Deallocation
    - malloc():# of byte
    - calloc() : # of object
    - realloc(): change size
    - free()



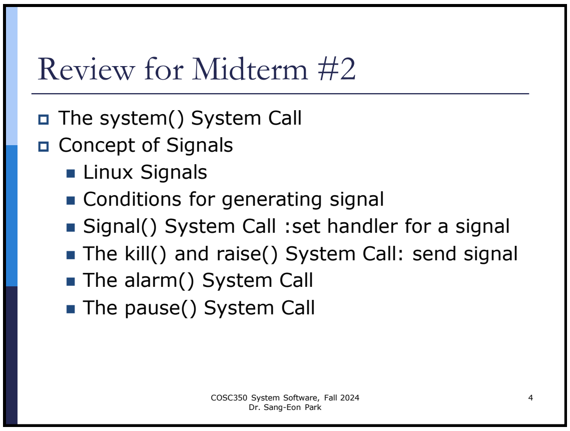
Review for Midterm #2

- Process Control
  - What is process? – process table
  - Process states – ready, running, block
  - Process identifier – 0: scheduler, 1:systemd
  - The fork(), vfork() System Call
  - File Sharing
  - Race Condition



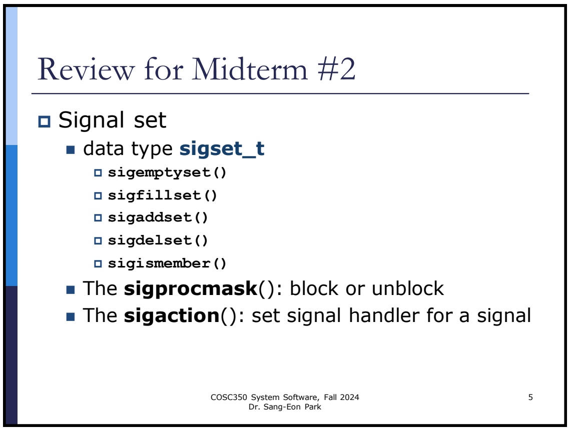
Review for Midterm #2

- Process Termination
  - Zombie Process: parent: live, child: dead
  - Orphan Process: parent: dead, child: live
- wait() and waitpid() System Call for process synchronization
- exec System Calls : a child can run different program



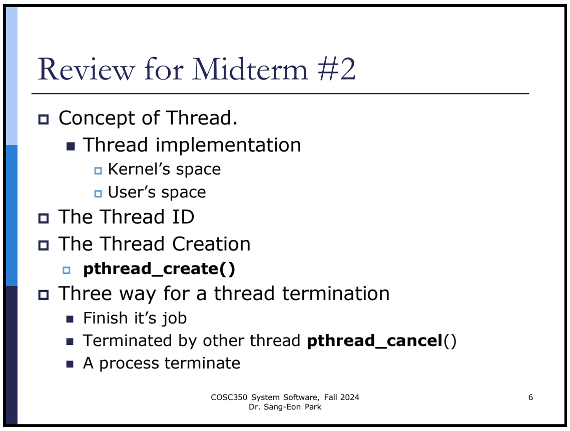
Review for Midterm #2

- The system() System Call
- Concept of Signals
  - Linux Signals
  - Conditions for generating signal
  - Signal() System Call :set handler for a signal
  - The kill() and raise() System Call: send signal
  - The alarm() System Call
  - The pause() System Call



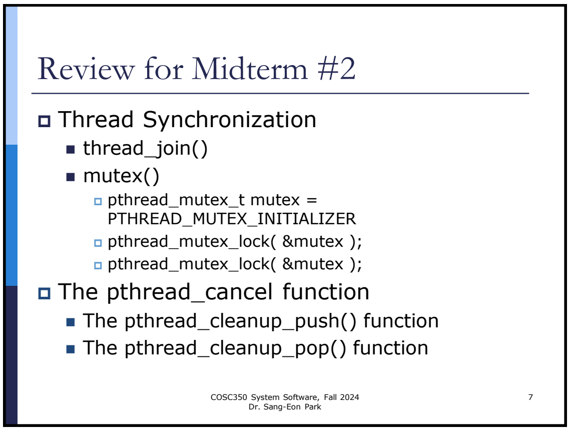
Review for Midterm #2

- Signal set
  - data type **sigset_t**
    - **sigemptyset()**
    - **sigfillset()**
    - **sigaddset()**
    - **sigdelset()**
    - **sigismember()**
  - The **sigprocmask**(): block or unblock
  - The **sigaction**(): set signal handler for a signal



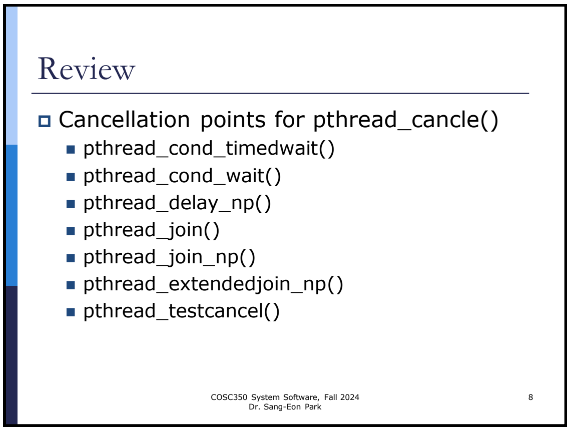
Review for Midterm #2

- Concept of Thread.
  - Thread implementation
    - Kernel’s space
    - User’s space
- The Thread ID
- The Thread Creation
  - **pthread_create()**
- Three way for a thread termination
  - Finish it’s job
  - Terminated by other thread **pthread_cancel**()
  - A process terminate



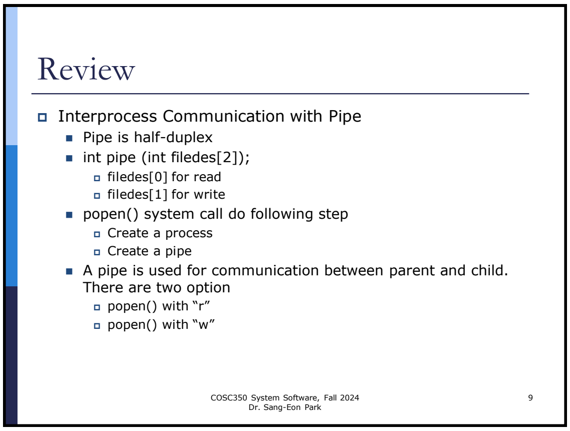
Review for Midterm #2

- Thread Synchronization
  - thread_join()
  - mutex()
    - pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER
    - pthread_mutex_lock(&mutex);
    - pthread_mutex_lock(&mutex);
- The pthread_cancel function
  - The pthread_cleanup_push() function
  - The pthread_cleanup_pop() function



Review

- Cancellation points for pthread_cancle()
  - pthread_cond_timedwait()
  - pthread_cond_wait()
  - pthread_delay_np()
  - pthread_join()
  - pthread_join_np()
  - pthread_extendedjoin_np()
  - pthread_testcancel()



Review

- Interprocess Communication with Pipe
  - Pipe is half-duplex
  - int pipe (int filedes[2]);
    - filedes[0] for read
    - filedes[1] for write
  - popen() system call do following step
    - Create a process
    - Create a pipe
  - A pipe is used for communication between parent and child. There are two option
    - popen() with “r”
    - popen() with “w”