



Operating Systems

(Homework 1)

Multi-threaded programming

These lecture materials are modified from the lecture notes written by A. Silberschatz, P. Galvin and G. Gagne.

Spring, 2018



Outline

- 1. Objectives
- 2. How to write a program ?
- 3. First program
- 4. Second program
- 5. Schedule



Objectives

- Writing two multi-threaded programs using the following functions;
 - pthread_create
 - pthread_join
 - pthread_exit



How to write a program ?

- POSIX thread programming (1)

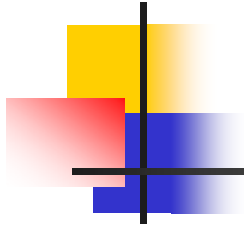
- Thread creation

- 1> Prototype

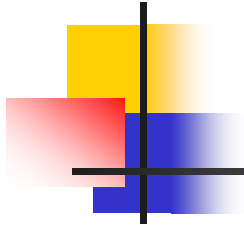
- `#include <pthread.h>`
 - `int pthread_create(pthread_t* tid, pthread_attr_t *attr, (void *) f, void *arg);`

- Roles

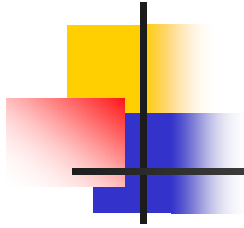
- Creates a new thread and runs the thread routine `f` with an input argument of `arg`
 - When `pthread_create` returns, argument `tid` contains the ID of the newly created thread



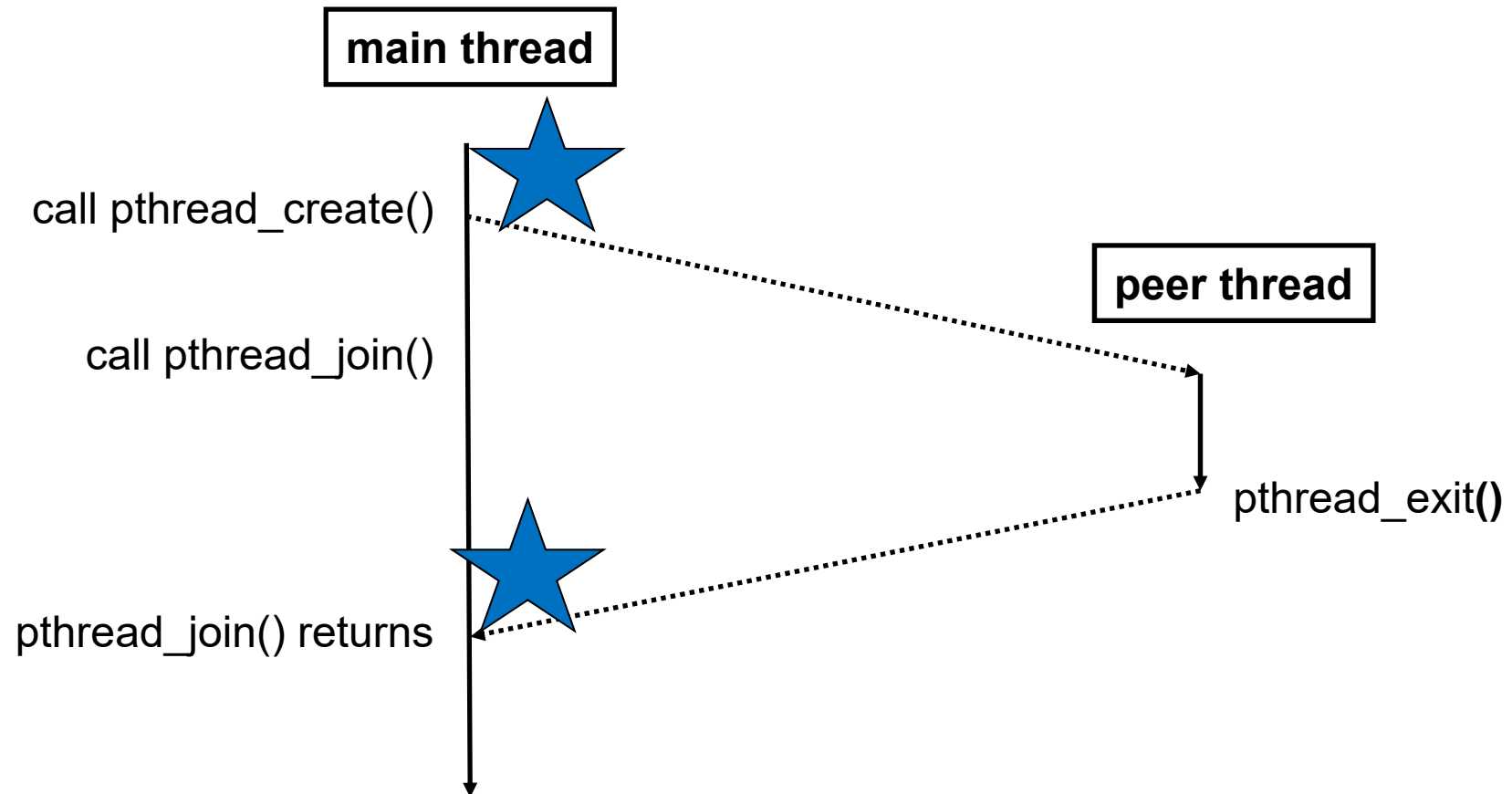
- POSIX thread programming (2)
 - Terminating the threads
 - Prototype
 - `#include <pthread.h>`
 - `int pthread_exit(void *thread_return);`
 - Roles
 - Terminating the thread with a return value of `thread_return` that will be transferred to `pthread_join`

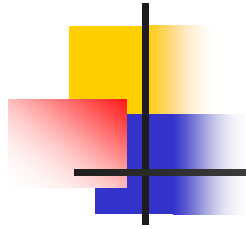


- POSIX thread programming (2)
 - Reaping terminated threads
 - Prototype
 - `#include <pthread.h>`
 - `int pthread_join(pthread_t tid, void *thread_return);`
 - Roles
 - `pthread_join` function blocks until thread `tid` terminates
 - It is similar to `wait` function but can only wait for a specific thread to terminate



S/W architecture

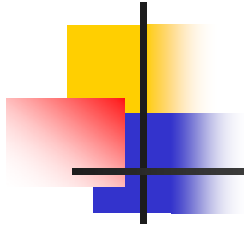




First program

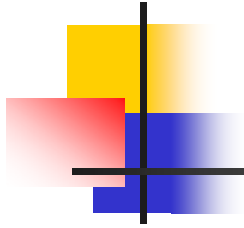
- Finding the highest value

| | | |
|-------|--|--|
| 5 3 4 | | |
| 6 7 2 | | |
| 1 9 8 | | |
| | | |
| | | |



■ Requirements

- 10개 thread 생성
 - Nine threads to check each of the 3 X 3 subgrids and returns the highest value to the tenth thread (selection thread)
 - Tenth thread (selection thread) selects one highest value from them (9 values)
 - Assumption
 - No same number
- 반드시 보고서에 10개의 쓰레드가 제대로 동작함을 기술할 것
- 모든 thread 는 가장 높은 수를 parent thread에게 return 함

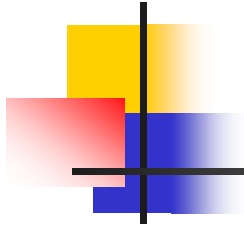


■ Requirements

- file로 숫자를 입력받을 것

input.txt

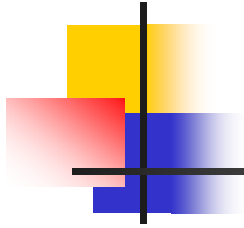
| | | |
|----------|----------|----------|
| 1 2 3 | 4 5 6 | 7 9 8 |
| 10 11 12 | 13 14 15 | 16 17 18 |
| 20 21 22 | 23 24 25 | 26 27 28 |
| ----- | ----- | ----- |
| ----- | ----- | ----- |
| ----- | ----- | ----- |
| ----- | ----- | ----- |
| ----- | ----- | ----- |
| ----- | ----- | ----- |



- 실행 예>

- highest input.txt
 - Highest number: 100

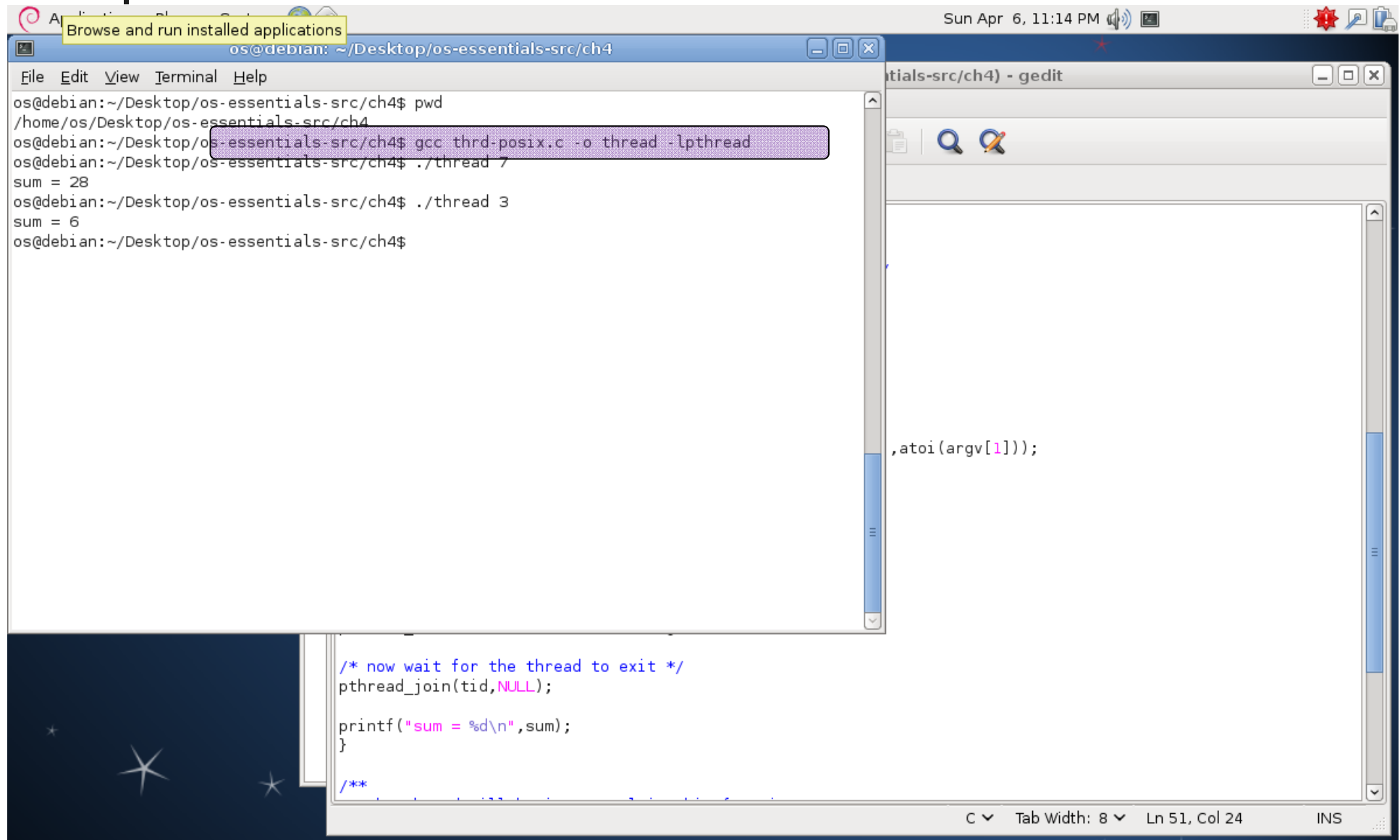
실행파일명



■ How to pass parameters to threads ?

```
int pthread_create( pthread_t* tid, pthread_attr_t *attr, (void *) f, void *arg);
```

<http://people.westminstercollege.edu/faculty/ggagne/osc/vm/index.html>



The screenshot shows a Linux desktop environment. At the top, there is a system bar with the date and time 'Sun Apr 6, 11:14 PM' and various system icons. Below the system bar, there are two windows. The left window is a terminal window titled 'os@debian: ~/Desktop/os-essentials-src/ch4'. It contains the following commands and output:

```
os@debian:~/Desktop/os-essentials-src/ch4$ pwd
/home/os/Desktop/os-essentials-src/ch4
os@debian:~/Desktop/os-essentials-src/ch4$ gcc thrd-posix.c -o thread -lpthread
os@debian:~/Desktop/os-essentials-src/ch4$ ./thread 7
sum = 28
os@debian:~/Desktop/os-essentials-src/ch4$ ./thread 3
sum = 6
os@debian:~/Desktop/os-essentials-src/ch4$
```

The right window is a gedit editor window titled 'os-essentials-src/ch4) - gedit'. It shows a C program snippet:

```
/* now wait for the thread to exit */
pthread_join(tid,NULL);

printf("sum = %d\n",sum);
}

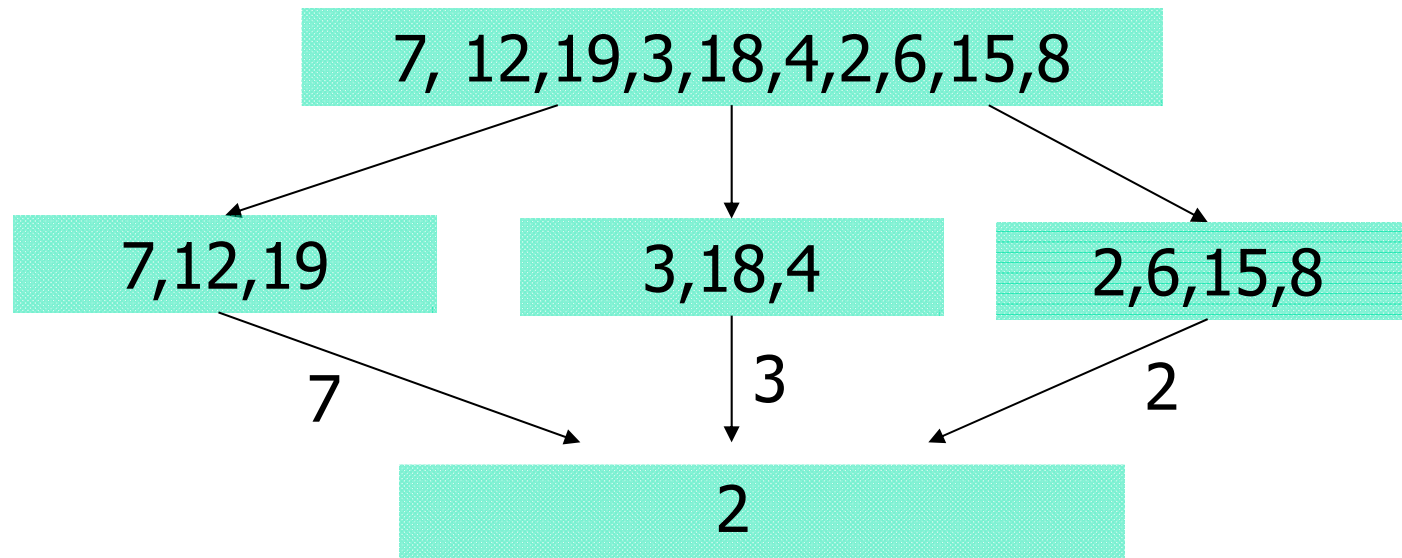
/**
```

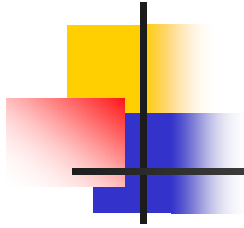
At the bottom of the screen, there is a status bar with the text 'C Tab Width: 8 Ln 51, Col 24 INS'.



Second program

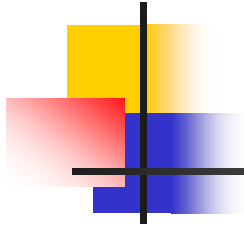
- Finding the lowest value
 - 1. The global array is shared by each thread
 - 2. The programming project requires passing parameters to each of the sorting threads
 - Starting index, Ending index





■ Requirements

- The merging thread must wait until three sorting threads calculate their results
 - Use a spin lock (busy looping) mechanism



- Input.txt

7 12 19 3 18 4 2 6 15 8

- 프로그램 실행 결과

- Lowest: 2



Schedule

- 데드라인

- 데드라인: 4월 19일 오후 23:59
 - 실제로는 4월 18일 오후 23:59 임
- 이후에는 어떤 경우에도 안 받으니, 하루 전에 미리 제출할 것을 권유함

- What to submit ?

- 보고서
 - 별도 평가 예정이니, 자세히 쓸 것
- source file, 출력 image