



OPERATING SYSTEMS

Programming examples on POSIX Threads

Chandravva Hebbi

Department of Computer Science

OPERATING SYSTEMS

Slides Credits for all PPTs of this course



- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:
 1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
 2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
 3. Some presentation transcripts from A. Frank – P. Weisberg
 4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

OPERATING SYSTEMS

- Programming examples on POSIX Threads

Chandravva Hebbi

Department of Computer Science

Thread creation

```
#include<pthread.h>
```

```
int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void  
*(*start_routine)(void*), void *arg);
```

Thread Completion

```
int pthread_join(pthread_t thread, void **value_ptr);
```

Locks

```
int pthread_mutex_lock(pthread_mutex_t *mutex);
```

```
int pthread_mutex_unlock(pthread_mutex_t *mutex);
```

When you have a region of code that is a critical section, and thus needs to be protected to ensure correct operation, locks are quite useful.

Example:

```
pthread_mutex_t lock;  
pthread_mutex_lock(&lock);  
x = x + 1; // or whatever your critical section is  
pthread_mutex_unlock(&lock);
```

Condition Variables

- The other major component of any threads library, and certainly the case with POSIX threads, is the presence of a condition variable.
- Condition variables are useful when some kind of signaling must take place between threads, if one thread is waiting for another to do something before it can continue.

Two primary routines are used by programs wishing to interact in this way:

```
int pthread_cond_wait(pthread_cond_t *cond, pthread_mutex_t *mutex);  
int pthread_cond_signal(pthread_cond_t *cond);
```

Demonstration of,

1. Write program to find sum of N natural numbers. Thread1 computes sum of even numbers(evensum) and thread 2 computes the sum of odd numbers (oddsum) and main thread computes the total sum.

Example: N=10

Even numbers=2,4,6,8,10

Odd numbers=1,3,5,7,9

Thread 1 result, Evensum=30

Thread 2 result, Oddsum=25

Main thread result=oddsum + evensum=30+25=55.

OPERATING SYSTEMS

Programming examples on POSIX Threads



2. Write a program to demonstrate producer consumer problem.
3. Write a program to demonstrate the dining philosopher's problem
4. Write a program create 2 threads, thread 1 prints lower case characters and thread 2 prints upper case character alternatively.

Note: synchronize the two threads using wait and signal

Expected output: aAbBcCdD.....zZ



THANK YOU

Chandravva Hebbi

Department of Computer Science Engineering

chandravvahebbi@pes.edu