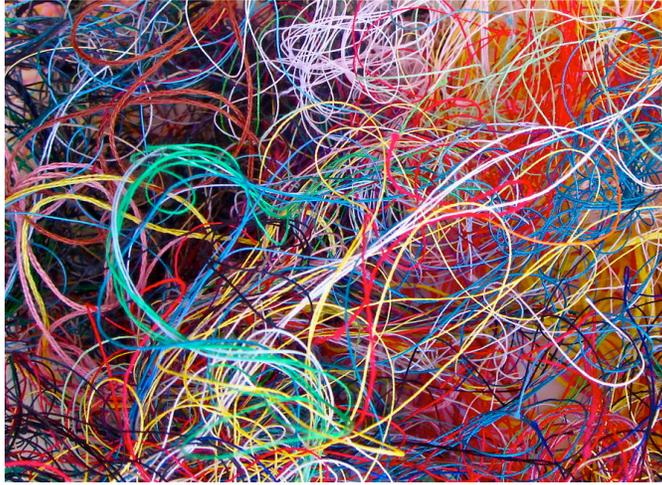
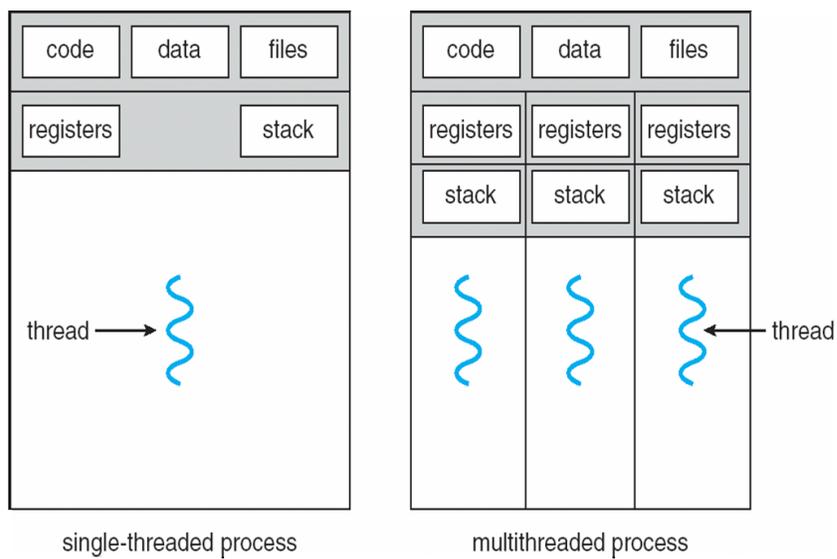


Threads



Multithreaded Processes



Thread APIs

POSIX Threads (Pthreads):

```
pthread_create(&tid, NULL, my_fun, &param); // thread runs my_fun
```

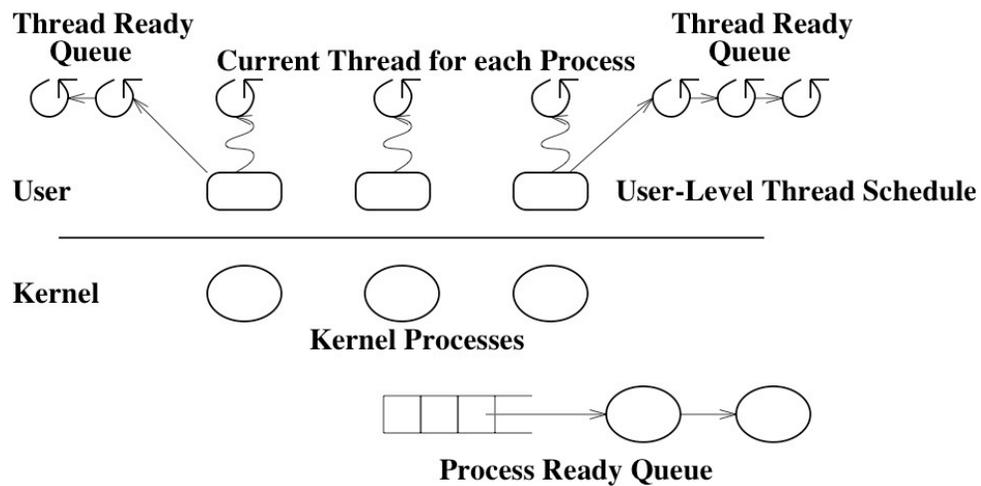
Win32 Threads:

```
ThreadHandle = CreateThread(NULL, 0, MyFun, &Param, 0, &ThreadID);
```

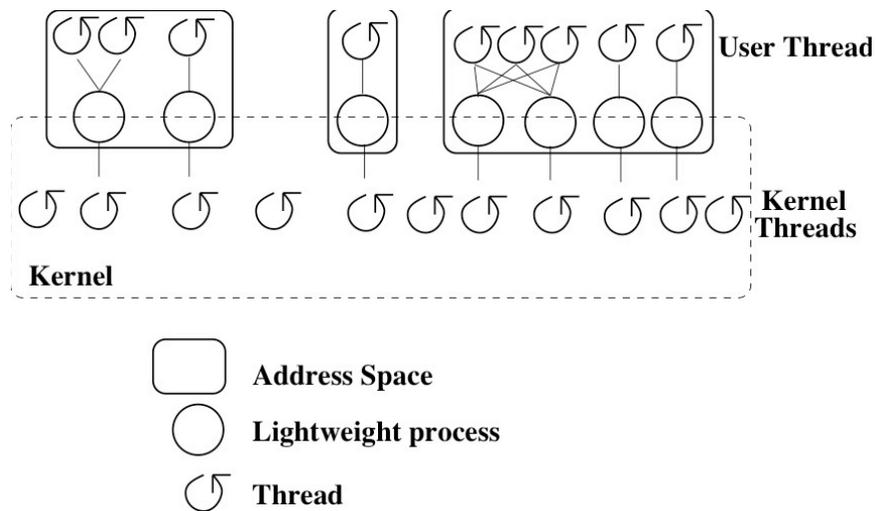
Java Threads:

```
Thread t = new Thread(new MyRunnable(param));  
t.start(); // start the thread running MyRunnable.run()
```

User-Level Threads



Lightweight Processes



Thread APIs

POSIX Threads (Pthreads):

```
pthread_create(&tid, NULL, my_fun, &param); // thread runs my_fun
```

Win32 Threads:

```
ThreadHandle = CreateThread(NULL, 0, MyFun, &Param, 0, &ThreadID);
```

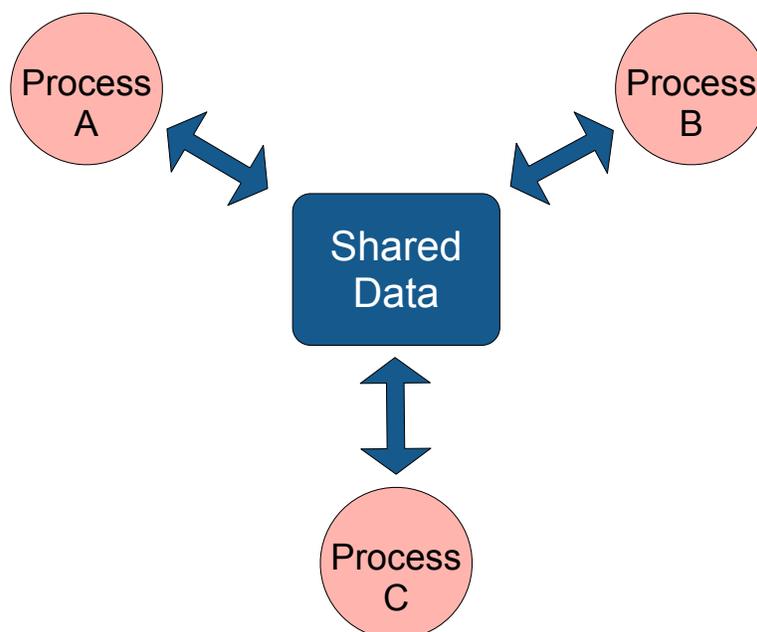
Java Threads:

```
Thread t = new Thread(new MyRunnable(param));  
t.start(); // start the thread running MyRunnable.run()
```

Threads Summary

- Threads vs processes
- User-level and kernel-level threads
- Thread interfaces vs implementations

Process Synchronization



ATM Example

```
// get funds in account
int total = get_account_total();

// get amount to withdraw
int withdraw = get_withdraw_amount();

// check for sufficient funds
if (total >= withdraw) {

    // withdraw funds from account
    update_account_total(total - withdraw);
    dispense_money(withdraw);

}
```

Too Much Milk

Time	You	Your Roommate
3:00	Arrive home	
3:05	Look in fridge, no milk	
3:10	Leave for grocery store	
3:15		Arrive home
3:20	Arrive at grocery store	Look in fridge, no milk
3:25	Buy milk	Leave for grocery store
3:35	Arrive home, put milk in fridge	
3:45		Buy milk
3:50		Arrive home with milk
3:50		Too much milk!

```
if (noMilk) {
    buy milk;
}
```

```
if (noMilk) {
    buy milk;
}
```

Too Much Milk: Solution 1?

Thread A

```
if (noMilk & NoNote) {  
    leave note;  
    buy milk;  
    remove note;  
}
```

Thread B

```
if (noMilk & NoNote) {  
    leave note;  
    buy milk;  
    remove note;  
}
```

Too Much Milk: Solution 2?

Thread A

```
leave note A;  
if (noNote B) {  
    if (noMilk) {  
        buy milk;  
    }  
}  
remove note A;
```

Thread B

```
leave note B;  
if (noNote A) {  
    if (noMilk) {  
        buy milk;  
    }  
}  
remove note B;
```

Too Much Milk: Solution 3?

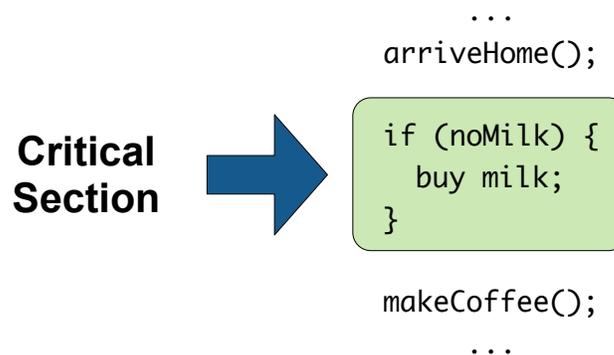
Thread A

```
leave note A;
while (note B) {
    do nothing;
}
if (noMilk) {
    buy milk;
}
remove note A;
```

Thread B

```
leave note B;
if (noNote A) {
    if (noMilk) {
        buy milk;
    }
}
remove note B;
```

Critical Sections



Requirement of **mutual exclusion**