

CPU Scheduling, Continued

Priority

One queue per priority level with FCFS used at each level.

Different algorithms can be used at different levels.

Very common.

Round Robin

A process runs for a Quantum—a fixed time unit. If the burst is consumed, the process is done. Otherwise it is forced back into the ready queue.

Multilevel Queues

Processes are classified by type, and prioritized by type.

Multilevel Feedback Queues.

Same as above, but processes can change type dynamically. Usually based on resource usage.

Shortest Remaining Time First

When we have different arrival times, and we're doing SJF, do we reschedule the current process when a new process arrives, or do we continue with the current process.

Multiprocessor Scheduling.

Surprisingly simple.

Threading

Threads are “cheap processes” almost always share memory with other threads.

There are Kernel Threads and User Threads.

Kernel Threads – Created and managed by OS.

User Threads – Created and managed by user – usually with a threading package.

Threading models

Many-to-One	All threading Package
One-to-One	All OS
Many-to-Many	Mixed

Pthreads

<http://www.llnl.gov/computing/tutorials/pthreads/>

This is the best on-line resource for pthreads information

Creating threads:

```
pthread_create (thread,attr,start_routine,arg)
```

```
pthread_exit (status)
```

```
pthread_attr_init (attr)
```

```
pthread_attr_destroy (attr)
```

```
pthread_join();
```

Mutex Variables

```
pthread_mutex_t mymutex = PTHREAD_MUTEX_INITIALIZER;
```

```
pthread_mutex_init (mutex,attr)
```

```
pthread_mutex_destroy (mutex)
```

```
pthread_mutexattr_init (attr)
```

```
pthread_mutexattr_destroy (attr)
```

```
pthread_mutex_lock (mutex)
```

```
pthread_mutex_trylock (mutex)
```

```
pthread_mutex_unlock (mutex)
```

Condition Variables

```
pthread_cond_t myconvar = PTHREAD_COND_INITIALIZER;
```

```
pthread_cond_init (condition,attr)
```

```
pthread_cond_destroy (condition)
```

```
pthread_condattr_init (attr)
```

```
pthread_condattr_destroy (attr)
```

```
pthread_cond_wait (condition,mutex)
```

```
pthread_cond_signal (condition)
```