pthreads

CS 241

Sept. 30, 2013

MP4

- Released on svn and website!
  - **Difficulty**: One-week MP
  - **Time**: Almost two weeks since the midterm is just *two weeks away*!

Thread Libraries

- CS 225:
  - **OpenMP**: Uses language constructs for creating multi-threaded code. Programmer never "creates" threads.

- CS 241:
  - **pthreads**: Low-level threading library. No automatic threading; a thread only does what you explicitly tell it to do.

Creating a Thread

```
pthread_create(
  pthread_t *tid,
  NULL,
  void *(*start_routine (void *)),
  void *arg
);
```

Example #1

```
void *thread(void *ptr) {
  printf("Hello World!");
  return NULL;
}

void main() {
  pthread_t tid;
  pthread_crerate(&tid, NULL, thread, NULL);



}
```

Exiting a Process or Thread?

- A **process** will exit when:
  1.
  2.
  3.
  4.

- A **thread** will exit when:
  1.
  2.

Waiting for a Thread

```
pthread_join(
  pthread_t tid,
  void **retval
);
```

```
void *add(void *ptr) {
   int i = *((int *)ptr);
   int result = i + 1;
   return &result;
}

void main() {
  pthread_t tid;
  int val = 2;
  pthread_crerate(&tid, NULL, thread, &val);

  int result;
  pthread_join(tid, &result);
  printf("add(%d) == %d\n", val, result);
}
```

```
void *compute(void *ptr) {
  char *s = (char *)ptr;
  printf("Line length: %d\n", strlen(s));
  return NULL;
}

void main() {
  size_t line_size = 100;
  char *line = malloc(line_size);
  FILE *file = fopen("strings.txt", "r");



  while ( getline(&line, &line_size, file) ) {
    pthread_t tid;
    pthread_create(&tid, NULL, compute, line);
  }


  close(file);
}
```

```
int ct = 0;
int X = 10000000;

void *up(void *ptr) {
  int i;
  for (i = 0; i < X; i++)
    ct++;
}

void *down(void *ptr) {
  int i;
  for (i = 0; i < X; i++)
    ct--;
}

void main() {
  pthread_t t1, t2;
  pthread_create(&t1, NULL, up, NULL);
  pthread_create(&t1, NULL, down, NULL);


  printf("Count: %d\n", ct);
}
```