Administrivia

- Reading assignments for this week posted (belatedly). Sample programs from class to be on Web soon.

- “Useful links” Web page has links to MPI and OpenMP sites. Go there to find complete documentation (standard/specification).
  (Some of it’s heavy going. Skim as you would man pages. OpenMP spec has many examples; MPI standard has some.)

Multithreaded Programming with OpenMP — Review

- Basic idea — fork/join programming model, all threads share memory.

- Can duplicate code in all threads (parallel directive), split a loop among threads (parallel for), have different threads do different things (parallel sections).
  More details in specification — can combine these in various ways.
  Various ways to assign loop iterations to threads — later.
Data Environment Clauses — Review/More

- Most variables are shared by default; exceptions are variables local to a block within a parallel region.

- To give each thread a separate copy — private clause. firstprivate and lastprivate can be used to start/end with shared value.

- To create a partial result in each thread and then combine (“reduce”) — reduction clause. Operations include sum, product, and/or. No max or min in C/C++.

Library Functions

- omp_get_num_threads, omp_set_num_threads, omp_get_thread_num — as in examples and appendix.

- omp_get_wtime — as in examples and appendix.

- Functions to do locking — later.

- Functions to do other things — in specification.
Synchronization Constructs

- **critical** — only one thread at a time executes this block of code. (Example — synch-2.c on sample programs page.)
- **barrier** — threads wait here until all have arrived. Implicit barrier at end of parallel region.
- **single** — only one thread executes this block.
- **Several others** — atomic, flush, ordered, master. More about them in the specification.

Locks

- **omp_lock_t** — declares a lock variable.
- **omp_init_lock, omp_destroy_lock** — create and destroy.
- **omp_set_lock** — acquire lock (wait if necessary).
- **omp_unset_lock** — release lock.
- **Other functions** described in specification.
- **Example** — synch-3.c on sample programs page.
Assigning Work to Threads — schedule clause

- **static** (with optional chunk size) — divide iterations into fixed-size blocks, distribute evenly among threads.
- **dynamic** (with optional chunk size) — queue of iterations, threads grab blocks of iterations until all done.
- **guided** (with optional chunk size) — like dynamic, but with decreasing blocks of iterations.
- **runtime** — get from OMP_SCHEDULE environment variable.

Intermezzo — Environment Variables (in **bash**)

- To set environment variable `FOO` for the rest of the session:
  ```bash
  export FOO=fooval
  ```
  (To set every time you log in, put in `.bash_profile`.)
- To run `bar` with a value for `FOO`:
  ```bash
  FOO=fooval bar
  ```
Numerical Integration Example, Revisited

- Last time we looked at two ways to parallelize the numerical integration example with OpenMP.

- One version used a `parallel for`; it performed well (nearly perfect speedup).
  (Try this again with various schedules.)

- The other version used a parallel region to parallelize as with MPI ("SPMD" model). Its performance was terrible. Let's try to figure out what's wrong ...(num-int-par-spmd-* .c on sample programs page).

Minute Essay

- None — sign in.