Shared Memory Multiprogrmming with Open MP:

- OpenMP is an API for shared-memory parallel programming. The “MP” in OpenMP stands for “multiprocessing,” a term that is synonymous with shared-memory parallel computing.
- OpenMP is designed for systems in which each thread or process can potentially have access to all available memory, and, when we’re programming with OpenMP, we view our system as a collection of cores or CPUs, all of which have access to main memory,

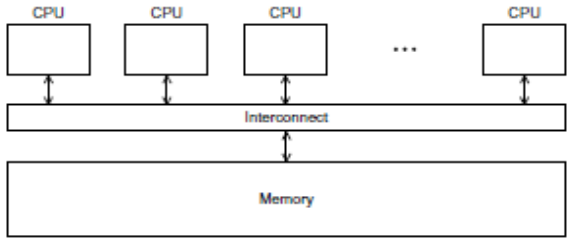

Fig :Shared Memory System

- OpenMP and Pthreads are both APIs for shared-memory programming.
There are few differences between tPthreads and OpenMP.

1) Pthreads requires that the programmer explicitly specify the behavior of each thread.

   OpenMP, allows the programmer to simply state a block of code should be executed in parallel, and the precise determination of the tasks and which thread should executethem is left to the compiler and the run-time system.

2) Pthreads (like MPI) is a library of functions that can be linked to a C program, so any Pthreads program can be used with any C compiler, provided the system has a Pthreads library.

   OpenMP, requires compiler support for some operations, and hence it’s possible that you may run across a C compiler that can’t compile OpenMP programs into parallel programs.

3) Pthreads is lower level and provides us with the power to program virtually any conceivable thread behavior. This has some associated cost—it’s up to us to specify every detail of the behavior of each thread.

   OpenMP, on the other hand, allows the compiler and run-time system to determinesome of the details of thread behavior, so it can be simpler to code some parallelbehaviors using OpenMP. The cost is that some low-level thread interactions can bemore difficult to program.