Husain Alshehhi

Scheduler was implemented using linked list. By adding a item, it appends it to the end of the list.

Scheduler::Scheduler
Initializes the values, mainly the head of the linked list.

Scheduler::add
Simply finds the end of the list and put the variable there.

Scheduler::yield
Takes the first element in the linked list. Pops that element from the list and runs it.

Schedule::resume
Similar to Scheduler::add

Scheduler::terminate
Does nothing interesting beside calling Scheduler::yield

thread_shutdown
Moved from thread.C to Scheduler.C. It calls the function Scheduler::terminate

thread termination:
Implemented by deleting the allocated memory and then terminate.

Issues with the implementation.
One of the problems was maintaining whether or not the linked list is empty (it sounds easier that it actually is). Another one is the Scheduler does not control the thread from the beginning (one of the threads runs without permission from the Scheduler) which renders the class into a non-efficient Scheduler.