main(argc,argv)

DEBUG('t', "Entering main")

(DebugIsEnabled(flag)

Intialize(argc,argv)

DebugInit(debugArgs)

Statstics()

Interrupt()

List()

Sceduler()

List()

Thread(“main”)

currentThread->setStatus(RUNNING)

interrupt->Enable()

SetLevel(IntOn)

ChangeLevel(old,now)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

OneTick()

DEBUG('i', "\n== Tick %d ==\n", stats->totalTicks)

DebugIsEnabled(flag)

ChangeLevel(IntOn,IntOff)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

CheckIfDue(False)

DebugIsEnabled(‘i’)

SortedRemove(&when)

ChangeLevel(IntOff,IntOn)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

CallOnUserAbort(CleanUp)

ThreadTest()

ThreadTest1()

DEBUG('t', "Entering ThreadTest1")

DebugIsEnabled(flag)

Thread(“forked thread”)

t->Fork(SimpleThread,1)

DEBUG('t', "Forking thread \"%s\" with func = 0x%x, arg = %d\n", name, (int) func, arg)

DebugIsEnabled(flag)

StackAllocate(func,arg)

AllocBoundedArray(StackSize \* sizeof(int))

Interrupt->setlevel(IntOff)

ChangeLevel(old,now)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

OneTick()

DEBUG('i', "\n== Tick %d ==\n", stats->totalTicks)

DebugIsEnabled(flag)

ChangeLevel(IntOn,IntOff)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

CheckIfDue(False)

DebugIsEnabled(‘i’)

SortedRemove(&when)

ChangeLevel(IntOff,IntOn)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

Scheduler->readytorun(this)

Thread->getName()

DEBUG('t', "Putting thread %s on ready list.\n", thread->getName())

DebugIsEnabled(flag)

thread->setStatus(READY)

readyList->Append((void \*)thread)

ListElement(item, 0)

IsEmpty()

interrupt->SetLevel(oldLevel)

ChangeLevel(old,now)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

OneTick()

DEBUG('i', "\n== Tick %d ==\n", stats->totalTicks)

DebugIsEnabled(flag)

ChangeLevel(IntOn,IntOff)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

CheckIfDue(False)

DebugIsEnabled(‘i’)

SortedRemove(&when)

ChangeLevel(IntOff,IntOn)

Debug(i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

SimpleThread(1)

Yield()

ChangeLevel(old,now)

DEBUG('i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

getName()

FindNextToRun ()

Remove()

SortedRemove(NULL)

IsEmpty()

ReadyToRun()

getname()

DEBUG('t', "Putting thread %s on ready list.\n", thread->getName())

DebugIsEnabled(flag)

setStatus(READY)

Append((void \*)thread)

ListElement(item, 0)

IsEmpty()

(back to Yield())

Run(nextThread)

CheckOverflow()

setStatus(RUNNING)

getName()

DEBUG('t', "Switching from thread \"%s\" to thread \"%s\"\n",

oldThread->getName(), nextThread->getName()

DebugIsEnabled(flag)

getName()

DEBUG('t', "Now in thread \"%s\"\n", currentThread->getName())

DebugIsEnabled(flag)

(back to Yield)

SetLevel(level)

ChangeLevel(old,now)

DEBUG('i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

OneTick()

DEBUG('i', "\n== Tick %d ==\n", stats->totalTicks)

DebugIsEnabled(flag)

ChangeLevel(IntOn, IntOff)

DEBUG('i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

CheckIfDue(False)

DebugisEnabled(‘I’)

SortedRemove(&when)

IsEmpty()

ChangeLevel(IntOff, IntOn)

DEBUG('i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

(back to yield)

(back to SimpleThread)

(Yield will run 5 times due to for loop)

(back to ThreadTest1)

(back to ThreadTest)

(back to main)

Finish()

SetLevel(IntOff)

ChangeLevel(old,now)

DEBUG('i',"\tinterrupts: %s -> %s\n",intLevelNames[old],intLevelNames[now])

DebugIsEnabled(flag)

(back to Finish)

getName()

DEBUG('t', "Finishing thread \"%s\"\n", getName())

DebugIsEnabled(flag)

Sleep()

getLevel()

getName()

DEBUG('t', "Sleeping thread \"%s\"\n", getName())

DebugIsEnabled(flag)

FindNextToRun ()

Remove()

SortedRemove(NULL)

IsEmpty()

Run(nextThread)

CheckOverflow()

setStatus(RUNNING)

getName()

DEBUG('t', "Switching from thread \"%s\" to thread \"%s\"\n",

oldThread->getName(), nextThread->getName()

DebugIsEnabled(flag)

getName()

DEBUG('t', "Now in thread \"%s\"\n", currentThread->getName())

DebugIsEnabled(flag)

(back to Run)

(back to Sleep)

(back to Finish)

(back to main)