```
 1 #include "syscall.h"
 2 #include "ipt.h"
 3 #include "system.h"
 4 #include "thread.h"
 5
 6 //----------------------------------------------------------------------
 7 // hashIpt
 8 //      Function to hash into IptHashTable.  Take a virtual page number
 9 // and process Id.  Returns an IptEntry.
10 //----------------------------------------------------------------------
11
12 IptEntry *hashIPT(unsigned int vpn, SpaceId id) {
13   return iptHashTable[(vpn+(id*PRIMESIZE))%IPT_HASH_TABLE_SIZE].entries;
14 }
15
16 //----------------------------------------------------------------------
17 // IptEntry::IptEntry
18 //      Inits data.
19 //----------------------------------------------------------------------
20
21 IptEntry::IptEntry(int vpnArg, int phyPageArg, IptEntry *prevIptArg)
22 {
23   pid=currentThread->pid;
24   vPage=vpnArg;
25   phyPage=phyPageArg;
26   prev=prevIptArg;
27   next=0;
28 }
29
30 //----------------------------------------------------------------------
31 // IptEntry::~IptEntry
32 //      Removes from list.
33 //----------------------------------------------------------------------
34
35 IptEntry::~IptEntry(void)
36 {
37   if(prev)
38     prev->next=next;
39   if(next)
40     next->prev=prev;
41 }
42
43 //----------------------------------------------------------------------
44 // IptHashTable::IptHashTable
45 //      Setup the sentinel
46 //----------------------------------------------------------------------
47
48 IptHashTable::IptHashTable(void)
49 {
50   entries=new IptEntry(0, 0, 0); // sentinel
51 }
52
53 //----------------------------------------------------------------------
54 // IptHashTable::~IptHashTable
55 //      Remove entries
```

```
 56 //----------------------------------------------------------------------
 57
 58 IptHashTable::~IptHashTable(void)
 59 {
 60   while(entries->next)
 61     delete entries->next;
 62   delete entries;
 63 }
 64
 65 //----------------------------------------------------------------------
 66 // MemoryTable::MemoryTable
 67 //      set invalid
 68 //----------------------------------------------------------------------
 69
 70 MemoryTable::MemoryTable(void)
 71 {
 72   valid=FALSE;
 73 }
 74
 75 //----------------------------------------------------------------------
 76 // MemoryTable::~MemoryTable
 77 //      null body
 78 //----------------------------------------------------------------------
 79
 80 MemoryTable::~MemoryTable(void)
 81 {
 82   // null body
 83 }
 84
```