```
File - /Users/JH/Documents/GitHub/NTU_OperatingSys_Lab/nachos-3.4/lab1/main.cc
 1 // main.cc
 2 // Bootstrap code to initialize the operating system kernel.
 3 //
 4 // Allows direct calls into internal operating system functions,
 5 //
       to simplify debugging and testing. In practice, the
 6 //
       bootstrap code would just initialize data structures,
 7 //
       and start a user program to print the login prompt.
 8 //
 9 // Most of this file is not needed until later assignments.
10 //
11 // Usage: nachos -d <debuaflags> -rs <random seed #>
12 //
           -s -x <nachos file> -c <consoleIn> <consoleOut>
13 //
           -f -cp <unix file> <nachos file>
14 //
           -p <nachos file> -r <nachos file> -l -D -t
15 //
                   -n <network reliability> -e <network orderability>
                   -m <machine id>
16 //
17 //
                   -o <other machine id>
18 //
                   -z
19 //
20 //
         -d causes certain debugging messages to be printed (cf. utility.h)
21 //
         -rs causes Yield to occur at random (but repeatable) spots
22 //
         -z prints the copyright message
23 //
24 //
       USER_PROGRAM
25 //
         -s causes user programs to be executed in single-step mode
26 //
         -x runs a user program
27 //
         -c tests the console
28 //
29 // FILESYS
30 //
         -f causes the physical disk to be formatted
31 //
         -cp copies a file from UNIX to Nachos
32 //
         -p prints a Nachos file to stdout
33 //
         -r removes a Nachos file from the file system
34 //
         -l lists the contents of the Nachos directory
35 //
         -D prints the contents of the entire file system
36 //
         -t tests the performance of the Nachos file system
37 //
       NETWORK
38 //
39 //
       -n sets the network reliability
40 //
        -e sets the network orderability
41 //
         -m sets this machine's host id (needed for the network)
42 //
         -o runs a simple test of the Nachos network software
43 //
44 //
       NOTE -- flags are ignored until the relevant assignment.
45 //
       Some of the flags are interpreted here; some in system.cc.
46 //
47 // Copyright (c) 1992-1993 The Regents of the University of California.
48 // All rights reserved. See copyright.h for copyright notice and limitation
49 // of liability and disclaimer of warranty provisions.
50
51 #define MAIN
52 #include "copyright.h"
53 #undef MAIN
54
```

55 #include "utility.h"

```
File - /Users/JH/Documents/GitHub/NTU_OperatingSys_Lab/nachos-3.4/lab1/main.cc
 56 #include "system.h"
 57
 58
 59 // External functions used by this file
 60 extern void ThreadTest(void), Copy(char *unixFile, char *nachosFile);
 61 extern void Print(char *file), PerformanceTest(void);
 62 extern void StartProcess(char *file), ConsoleTest(char *in, char *out);
 63 extern void MailTest(int networkID);
 64 extern void SynchTest(void);
 65
 66 //-----
 67 // main
 68 // Bootstrap the operating system kernel.
 69 //
 70 // Check command line arguments
 71 // Initialize data structures
 72 // (optionally) Call test procedure
 73 //
 74 //
       "argc" is the number of command line arguments (including the name
           of the command) -- ex: "nachos -d +" -> argc = 3
 75 //
 76 // "argv" is an array of strings, one for each command line argument
           ex: "nachos -d +" -> argv = {"nachos", "-d", "+"}
 77 //
 78 //-----
 79
 80 int
 81 main(int argc, char **argv)
 82 {
 83
        int argCount;
                               // the number of arguments
 84
                       // for a particular command
 85
        DEBUG('t', "Entering main");
 86
        (void) Initialize(argc, argv);
 87
 88
 89
        ThreadTest(); //Start ThreadTest();
 90
 91
        for (argc--, argv++; argc > 0; argc -= argCount, argv += argCount) {
 92
        argCount = 1;
 93
            if (!strcmp(*arqv, "-z"))
                                                  // print copyright
 94
               printf (copyright);
 95 #ifdef USER_PROGRAM
           if (!strcmp(*argv, "-x")) {
                                             // run a user program
 96
 97
            ASSERT(argc > 1);
                StartProcess(*(argv + 1));
 98
 99
                arqCount = 2;
            } else if (!strcmp(*argv, "-c")) { // test the console
100
            if (argc == 1)
101
                ConsoleTest(NULL, NULL);
102
103
            else {
            ASSERT(argc > 2);
104
               ConsoleTest(*(argv + 1), *(argv + 2));
105
106
               argCount = 3;
            }
107
108
            interrupt->Halt(); // once we start the console, then
                       // Nachos will loop forever waiting
109
110
                       // for console input
```

```
File - /Users/JH/Documents/GitHub/NTU OperatingSys Lab/nachos-3.4/lab1/main.cc
111
112 #endif // USER_PROGRAM
113 #ifdef FILESYS
        if (!strcmp(*argv, "-cp")) { // copy from UNIX to Nachos
114
115
            ASSERT(argc > 2);
            Copy(*(argv + 1), *(argv + 2));
116
117
            argCount = 3;
118
        } else if (!strcmp(*argv, "-p")) { // print a Nachos file
119
            ASSERT(argc > 1);
120
            Print(*(argv + 1));
121
            argCount = 2;
        } else if (!strcmp(*argv, "-r")) { // remove Nachos file
122
123
            ASSERT(argc > 1);
124
            fileSystem->Remove(*(argv + 1));
125
            araCount = 2;
        } else if (!strcmp(*argv, "-l")) { // list Nachos directory
126
                 fileSystem->List();
127
        } else if (!strcmp(*argv, "-D")) { // print entire filesystem
128
                fileSystem->Print();
129
        } else if (!strcmp(*argv, "-t")) { // performance test
130
131
                PerformanceTest();
132
        }
133 #endif // FILESYS
134 #ifdef NETWORK
135
            if (!strcmp(*argv, "-o")) {
            ASSERT(argc > 1);
136
137
                 Delay(2);
                                         // delay for 2 seconds
138
                             // to give the user time to
                             // start up another nachos
139
                MailTest(atoi(*(argv + 1)));
140
141
                argCount = 2;
142
            }
143 #endif // NETWORK
144
        }
145
                                  // NOTE: if the procedure "main"
146
        currentThread->Finish();
147
                     // returns, then the program "nachos"
                     // will exit (as any other normal program
148
149
                     // would). But there may be other
150
                     // threads on the ready list. We switch
151
                     // to those threads by saying that the
152
                     // "main" thread is finished, preventing
                     // it from returning.
153
154
                             // Not reached...
        return(0);
155 }
156
```