

1. (50 points) These questions are concerned with the Nachos operating system.

- a. Where in the code does Nachos decide whether or not to print a copyright notice? What does the user have to do to cause this notice to be printed?**

If the user runs Nachos by typing “nachos -z”, Nachos prints a copyright message, as documented in the comments at the beginning of main.cc. The code to test for the “-z” option is the line “if (!strcmp(*argv, “-z”))” in main().

- b. What is the relationship between main.cc and system.cc, and the major functions in each of those two files?**

The function main() in main.cc calls the Initialize() function in system.cc. Function main() tests for most command line arguments, runs different tests depending on how Nachos is compiled, and calls function Initialize(). Function Initialize() initializes Nachos’ various data structures, as well as the main thread, scheduler, interrupts, etc.

- c. In the function to de-allocate a thread, what does the ASSERT function do and why is it needed?**

In Thread::~Thread(), ASSERT is used to verify that the object being deallocated (“this”) is indeed the thread currently running (currentThrea). The function ASSERT (defined in utility.h) checks the specified condition, and if it is false, prints an “Assertion failed” message and drops a core file.

- d. What function does Nachos use to put a thread on the ready list to wait for CPU time, and what does that function do (explain in your own words)?**

Nachos uses the Scheduler::ReadyToRun() function, defined in scheduler.cc, to append a thread to the end of the ready list and change its state to “READY”.

- e. Is the list function Append a function built into C++? Explain.**

No, List::Append() is defined in list.cc, as part of the List class defined there.

2. (40 points) These questions are concerned with the emulated machine that runs underneath the Nachos operating system.

- a. How many registers (including stack pointer, program counter, etc.) does the CPU have, and where is that total defined?**

There are 40 registers, allocated as int registers[NumTotalRegs] in machine.h, where NumTotalRegs is defined as 40.

- b. What does the BLEZ instruction do (be specific, don’t just guess based on the name)?**

BLEZ is the “branch if less than or equal to zero” instruction. The PC normally points to the next instruction after the current one, but if the source register (pointed to by “rs”) is less than or equal to zero, BLEZ adds an offset index to the current PC value to produce a new PC value.

- c. What does Interrupt::Interrupt do? Would it be appropriate to call this function more than once? Explain.**

Interrupt::Interrupt initializes the emulated hardware to implement interrupts. Not that it does **not** initialize the interrupt handler, but only the emulated hardware. Since the CPU would normally only have one set of hardware to implement interrupts, it would be inappropriate to call this function more than once.

- d. In what data structure does Nachos keep track of the amount of time spent in user code versus code?**

In the file stats.h, the class Statistics keeps track of these two amounts as systemTicks and userTicks, respectively, where a Tick is a tick of the system clock.

3. (20 points) Compile and run Nachos and observe the output. The comments for ThreadTest say it sets up a “ping-pong” — what does this mean, and how do you observe it happening?

Output not shown here...

It's like a ping-pong ball going back and forth between two players of that game. Thread 0 runs, then thread 1 runs, and they keep taking turns one after the other. This is clearly shown in the program output.