CS 4/53201 # Project #1 Solution **Operating Systems**

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

   a. When you compile as instructed, and run Nachos with the "–D" switch, what happens? Why doesn't the file system get printed, as the comments in main.cc would seem to indicate?

   Nachos simply ignores the "–D" switch, not even printing out an error message.

   In main.cc, the "-D" argument (which should print out the contents of the file system) is only compiled if nachos is compiled using the "FILESYS" switch.

   As we can see below, when nachos is compiled as instructed, the "THREADS" switch is used (which is why ThreadTest() runs), but the "FILESYS" switch is not used.

   cd threads; /local/opt/gcc/bin/make depend g++ -I../threads -I../machine -DTHREADS -DHOST_SNAKE -DHOST_IS_BIG_ENDIAN -DCHANGED ...

   b. Where and how is the main thread in Nachos started?

   In system.cc, the main thread is created by the statement "currentThread = new Thread("main");".

   c. When a context switch occurs, where are a thread's registers stored? How big is this storage area, and where **in the code is that size defined?**

   The registers are stored in machineState, an array in the Thread object's private storage. It is defined to be an array of size 18 by the statement "#define MachineStateSize 18" in thread.h.

   d. What does Thread::Yield() do if no other thread is on the ready list?

   It restores the interrupt state and continues executing the current thread.

   e. What happens to the argument "arg" passed to Thread::Fork?

   It gets passed to the function which the forked thread is to execute.

2. **(30 points) These questions are concerned with the emulated machine that runs underneath the Nachos operating system.**

   a. Wh**at do the first 5 lines of Machine::Machine do?**

   Zero the machine registers, allocate space for memory, and initialize the memory space to all zeros.

   b. What is the difference between the MIPS "MULT" and "MULTU" instructions? How is this difference implemented?

   "MULT" does a signed multiply, while "MULTU" does an unsigned multiply. The argument that specifies whether the multiplication is to be signed or unsigned is passed to the "Mult" function at the end of mipssim.cc.

   c. What are the possible states for an interrupt, and where are those states defined?

   The states are "on" and "off", defined at the beginning of interrupt.cc.

3. **(20 points) Compile and run Nachos and observe the output. Then modify "ThreadTest" to fork a second thread "u" named "second forked thread" immediately before it forks thread "t" named "forked thread". What are the results when Nachos is compiled and run? What order do the threads run in, and why)?**

   Output not shown here…

   Thread 0 runs, then 2, then 1. 0 was running before, so it keeps running. 2 gets created and added to the end of the ready list, then the same for 1, so once 0 yields, 2 runs, then 1, and then back to 0 again.