ECS150
Operating Systems
Winter Quarter 2004

Lab 2
Process Management

(Due date and handin procedure to be posted later.)

These problems ask you to make very small modifications to Minix. You will be modifying the kernel, recompiling and assembling it using `MAKE` to link the various executable files, and then produce a new Minix boot disk. All of this is quite straightforward, as explained in Sophies posting to the website. Most of the code you will deal with is in `proc.c` but some in `clock.c` (the clock task).

Each of these programs requires only trivial modifications to Minix. The key is locating the current source code that should be modified; there are many acceptable solutions.

Note: In most of the problems you are asked to add fields to the process table, but only a subset of the processes will refer to these fields. Clearly, all process entries (for all processes) will have these additional fields.

1. Extend the process table with a field that records the time of creation of each user process, excluding `init`. Make sure you update this field upon process creation. When demonstrating your new kernel, you should provide convincing evidence that the recorded creation time is indeed plausible. You might use one of the currently allocated hot keys, e.g., `F1`, to, when struck, print out the relevant fields in the process table, but you will have to change the actions associated with `F1`.

2. This question asks you to record system calls and various messages, and to reconcile the values you have recorded. To be more specific, extend the process table to record (1) for each user process the number of messages it sends to either the file server or the memory manager, (2) for each user process the number of messages it receives from either the file server or the memory manager, (3) for each user process the number of system calls it causes. Reconcile the values you have recorded; for example have one of the programs make a call to `pause`. Write and execute one or more programs that will cause the number of recorded values to increase dramatically. Again, use a hot key to print out the relevant fields in the process table.

3. This question asks you to change the scheduling of user processes, in particular to have the quantum q be dynamically modified. In particular, the value of q is to be given by the following ratio:

$$\frac{\text{max response time}}{\text{number of user process in ready queue}}$$

   The maximum response time is to be itself dynamically adjustable, toggling between 2 and 3 seconds by striking a hot key of your choosing. It should be easy to demonstrate your new

scheduler: run a large number of processes that are executing programs that are in an infinite loop and run a process that is I/O bound in that it will only print a brief string every time through a loop. The time between actual prints is then the max response time.