Minix Scheduling

It is highly recommended that you start on this early (the earlier you run into problems, the earlier we can help you fix them).

The First Task

The goal of this first task is to make sure you are comfortable recompiling the minix source code. This portion of the assignment should be completed as soon as possible, but will not be graded separately.
Your goal in this task is to find the source file in the minix source code that prints the copyright message upon boot. You will add an extra print command to follow that copyright message. The copyright message that is printed is actually part of what is called the "MINIX startup banner" (this should help you locate the file).

The Two-Week Assignment

The goal of this assignment is to gain experience in modifying an operating system kernel, and specifically the process scheduling algorithm. Using the Minix 3 operating system, running under vmware, you are to modify the queue selection algorithm to skew the priority scheduling. Specifically, if the current selection is based on a pure priority order, then you are to include a random selection of a lower-level priority job. Note that if you modify the priority queue imprudently, the operating system will cease to function. Your goal is to achieve the following: upon attempting to select the next job, modify the selection to add a slight possibility of choosing from a different level (the lower the probability you choose for this, the more consistent with the current selection mechanism you are, some experimentation may be required to select a reasonable value). You will be graded based on both your
implementation of this modification, and on how well you explain the mechanism. It is therefore important to realize that this is both an exercise in coding, as well as an exercise in familiarizing yourself with, and understanding to the point of being able to explain, an unfamiliar code base.

The Deliverables

1. ***A write-up of your changes, your efforts to test them, and a description of the current MINIX scheduling algorithm:***

   This portion of the assignment is as critical, if not more so, than the actual implementation of your solution. In addition to describing the code you provide, a complete write-up will include a discussion of how you tested your solution. Give careful thought to this question, and be sure to describe how you ***would*** *test your code, even if a particular approach was impractical for you to execute on this particular assignment. In addition you should start with a description of the current Minix scheduler, referring to specific functions as part of your description.*

2. ***Modified source code:***

   This can be limited to a single file, in fact the most critical changes need only be applied to a single function within the Minix kernel. To be considered complete, the solution you provide must have switched from the current priority queue, to a randomized priority queue. You need to introduce randomness in selecting the next job to run. You are free to introduce such randomness in any manner you choose, but ideally you will maintain the goal of the existing scheduling algorithm to favor higher-priority queues over lower priority queues (by biasing the random selection towards higher-priority queues).