

3.2 quantum is defined in kernel.h which has the value of 2

3.2 a) INITPRIO is 20, so the process running main has the priority of 20. The console is printing a chunk of As, then a chunk of Bs, then a chunk of As and then a chunk of Bs, repeatedly and endlessly. Because sndch is a while(1) loop, so once main process with priority 20 is done, the only processes that is competing for cpu resources are the two processes we created here, who have the same priority 10 and will keep printing letters until its process is context switched out.

3.2 b) The console is still printing As and Bs with the same pattern, however each chunk is much smaller. In problem A, each chunk will span one and half lines. however, in problem B, each chunk is about a quarter of a line.

3.2 c) The console is only printing Bs now.

3.2 d) The console is only printing As now.

3.2 e) The console is also printing As, there is no difference in terms of output. However, since main and process A has the same priority, main process gets to execute and do its stuff (Create the shell, etc).

3.2 f) Because we need at least one ready process (whose stack can be used) to be always available and handle interrupts, when no other processes are running. Thus null process with the least priority 0 is necessary. If a process with zero or negative priority can be created, then we can no longer guarantee null process to be always available.

4 No written answer required.

5 No written answer required.