IN1011 Lecture 3 Exercises

Questions:

1. How are 73_x and 50_x written (in base x)?

2. What is the largest number that can be represented by 11 bits?

3. In base 10, we have $100 - 1 = 99$. Using this hint, what is $1000_{x+1} - 1$ in base $x + 1$?

4. What is $2^{11} - 1$ in base 2?

5. In a 7-state process model, how many transitions can be due to:

   (a) the long-term scheduler?

   (b) the medium-term scheduler?

   (c) the short-term scheduler?

   (d) the completion of an I/O request?

   (e) a program attempting to perform an illegal operation?

6. Some CPU scheduling policies incur more "overhead" than others; here, "overhead" means the time spent by the OS in deciding, and replacing, a process running on the CPU with another process waiting in the ready queue. Arrange the following CPU scheduling policies – RR, FCFS, SPN, and SRT – in the order of the overhead they produce, starting from shortest overhead to longest overhead. You may assume the same processes are scheduled by the different policies and that round-robin is using very small time-quanta.

Solutions:

1. Expand as a polynomial in powers of the base. So, $73_x = 7x^1 + 3x^0 = 7x + 3$ and $50_x = 5x^1 + 0x^0 = 5x$;

2. Each digit in an 11-bit word can take one of two possible values – i.e. either 0 or 1. So, 11 bits can represent 2^{11} different numbers. These numbers start at zero (i.e. all 11 digits are zero) then increase by 1 up until the largest value. So, the largest value must be $2^{11} - 1$ (i.e. the number of possibilities minus the zero representation). If the representations started at one instead of zero, then the largest number would be the number of possible representations, which is 2^{11}; but, because the representations start at zero, 2^{11} is an incorrect answer;

3. Using the hint, when we subtract 1 from a number that is a one followed by only zeroes, the result of this subtraction will be a number with the same number of digits as the zeroes, and each digit will be one-less than the base (e.g. 99 has two digits because there are two zeroes in 100, and each digit is one-less-than the base 10). So, $1000_{x+1} - 1 = xxx_{x+1}$;

4. To solve this, we bring together the previous answers. So,

$$2^{11} - 1 = 1\overbrace{00...00}^{11}{}_2 - 1 = \overbrace{11...11}^{11}{}_2 \, .$$

5. Refer to the 7-state process model drawn during the lecture, where I indicated the following transitions:

   (a) 2 transitions – from `new` state to either the `ready` state or `ready-suspend` state;

   (b) 4 transitions – between the `ready-suspend` and `ready` states, or between the `ready-blocked` and `blocked` states?

   (c) 2 transitions – between the `ready` and `running` states;

   (d) 2 transitions – from the `blocked` to the `ready` stae, and from the `blocked-suspend` to the `ready-suspend` state;

   (e) 2 possibilities – either the process is moved from the `running` to the `ready-suspend` state or, more typically, the process is terminated by moving from the `running` to the `exit` state;

6. FCFS and SPN both have non-preemptive decision modes, so these policies incur similar and less overhead than policies with preemptive decision modes like RR and SRT. Since we assume RR with very small time-quanta, the OS will have to change the process running on the CPU more often than SRT; because, under SRT, the OS will only change the running process if a process arrives in the ready queue with an estimated shorter remaining time to execute. So, either FCFS, SPN, SRT, RR or SPN, FCFS, SRT, RR, are acceptable answers.

Bonus questions:

1. In general, for $x \geqslant 1$, what is $(x+1)^n - 1$ in base $x+1$? Hint: guess the answer by trying out some numerical examples, and use previous answers in this worksheet for inspiration;

2. Suppose you are debugging code you have written and, by using a debugger in UNIX, you place "breakpoints" in the code. Assume the 7-state process model, and assume the code is executing when it reaches a breakpoint.

   (a) What state was your executing code in just-before the breakpoint?

   (b) What state is your executing code in immediately-after the breakpoint?

(c) explain why each of the 3 OS schedulers could NOT have caused this transition.

(d) If none of these schedulers caused this transition, give a plausible explanation of which process caused the transition and how it did so.