COMP3911 Suggested Exercises #2b

These suggested exercises are not to be handed in for marking, but should be used as a study aid.

1) What is the benefit of having separate "administrator" and "user" accounts? Give an example of an activity which, when done as "administrator", poses a risk to the system.

> *Most actions done on a computer do not require "administrator" priveledges. If these are done with an accout with full priveledges, the risk of "accidents" is greatly increased.*
>
> *Further, with the ability to provide trojans, worms and viruses over the web, it is much easier to compromise machines with over-use of the administrator account. If a running program is compromised, or turns out to be a trojan, it is allowed to do anything at all to your machine if it was running as administrator.*

2) Why is the POSIX standard important? What does this get us? What do standards in general buy us?

> *The POSIX standard allows us to write C code to interface with an operating system which should function the same on any operating system implementing the POSIX standard.*
>
> *In effect, the POSIX standard aids us in writing "for portability" among different operating systems.*
>
> *Standards in general are presented to allow communication between groups working independantly. In this case, the groups are the operating system implementors, and the implementors of code to run on the operating system.*

3) We can discount the memory as the bottleneck immediately, as if memory was the limiting factor there will be few if any free memory pages. If pages are unused now, then buying more RAM will simply mean increase the number of untouched pages, and will not affect the underlying problem.

Both the CPU and disk have queues, so both are amenable to examination using M/M/1. Whichever device has the longer average queue will be the bottleneck device.

If the CPU is idle 20% of the time, then for the CPU,

$\rho_{CPU} = 1 - P_0 = 1 - 0.2 = 0.8$

The disk is just slightly more complicated as we need to figure out how long an I/O access is. Converting 7200 RPM into time per rotation gives us:

7200RPM -> 120 rev/sec -> 8.33msec for 1 revolution

Adding in our estimated seek time of ~10msec gives us 18.33msec per I/O request. Since we have 10 I/O operations per second, we are spending about 183.3msec of each second (1000 msec), and we are idle the remaining time. We therefore calculate P_0 and/or ρ_{DISK} as:

$P_0 = 1 - 0.1833$ <-> $\rho_{DISK} = 0.1833$

We can now compare our two values for ρ, and as $\rho_{DISK} \ll \rho_{CPU}$, we can definitively state that buying a faster disk won't solve the immediate problem, as it is the CPU which is the bottleneck device.

If we double the CPU speed, we are doubling the service rate (μ), so when we adjust ρ_{CPU} accordingly we expect it to halve, reducing to 0.4.

At this point, the CPU is still the bottleneck device, however it will be able to handle an increase in λ of a factor of two before the same service rate is seen.

If the change in CPU speed caused ρ_{CPU} to become lower than ρ_{DISK}, then the disk would become the new bottleneck device, and further change in CPU without speeding up the disk would no longer reap a benefit.