Homework Assignment #5

Q1. Please discuss: 1) the difference between the memory-mapped I/O and the direct I/O (or I/O-mapped I/O); and 2) the advantages and disadvantages of the memory-mapped I/O.

ANSWER: Memory-mapped I/O maps all control registers into the memory space with a single unified address space. I/O mapped I/O (also known as direct I/O) uses a separate, dedicated address space and is accessed via a dedicated set of microprocessor instructions.

Advantages of memory-mapped I/O:

- Device control registers can be addressed in C/C++ (like other variables) thus programs can be written entirely in C/C++
- No special protection needed, OS simply refrains from putting the portion of address space containing control registers to any virtual address space
- Every instruction that can reference memory can be used to reference control registers.

Disadvantages of memory-mapped I/O:

- Caching is problematic, need to disable caching control registers
- With multiple buses, special measures need to be taken to make sure I/O devices see memory references (e.g. snooping)

Q2 (Problem 5.14). In which of the four I/O software layers is each of the following done.
(a) Computing the track, sector, and head for a disk read.
(b) Writing commands to the device registers.
(c) Checking to see if the user is permitted to use the device.
(d) Converting binary integers to ASCII for printing.

ANSWER:

(a) Device driver.
(b) Device driver.
(c) Device-independent software.
(d) User-level software.

Q3 (Problem 5.21). A RAID-5 can fail if two or more of its drives crash within a short time interval. Suppose that the probability of one drive crashing in a given hour is p. What is the probability of a k-drive RAID failing in a given hour?

ANSWER: The probability of 0 failures, P_0, is $(1 - p)^k$. The probability of 1 failure, P_1, is $kp(1-p)^{k-1}$. The probability of a RAID failure is then $1 - P_0 - P_1$. This is $1-(1-p)^k - kp(1-p)^{k-1}$.

Q4 (Problem 5.31). Disk requests come in to the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. A seek takes 6 msec per cylinder moved. How much seek time is needed for

(a) First-come, first served.
(b) Closest cylinder next (Shortest Seek First).
(c) Elevator algorithm (initially moving upward).
In all cases, the arm is initially at cylinder 20.

ANSWER:
(a) 10+12+2+18+38+34+32=146cylinders =876msec.
(b) 0+2+12+4+4+36+2 = 60cylinders =360msec.
(c) 0+2+16+2+30+4+4 = 58cylinders =348msec.

Q5 (Problem 5.53). If a CPU's maximum voltage, V, is cut to V/n, its power consumption drops to $1/n^2$ of its original value and its clock speed drops to l/n of its original value. Suppose that a user is typing at 1 char/sec, but the CPU time required to process each character is 100 msec. What is the optimal value of n and what is the corresponding energy saving in percent compared to not cutting the voltage? Assume that an idle CPU consumes no energy at all.

ANSWER: Since CPU can process one character in 100 msec, and user is typing at 1 char/sec, we can drop the CPU clock speed to 1 sec/100 msec=1/10 of its original value in this scenario, which means an optimal value of n=10 in this case.

If the energy consumed in 1 sec at full speed is *E*, then running at full speed for 100 msec then going idle for 900 msec uses *E*/10. Running at 1/10 speed for a whole second uses *E*/100, a saving of 9*E*/100. The percent savings by cutting the voltage is 90%.

Q6. Please briefly explain what happens in terms of the client, client stub, client's OS, server, server stub, server's OS in steps when an RPC (remote procedure call) is invoked?

ANSWER: When an RPC is invoked, the following steps will take place:
1. The client procedure calls the client stub in the normal way.
2. The client stub builds a message and calls the local operating system.
3. The client's OS sends the message to the remote OS.
4. The remote OS gives the message to the server stub.
5. The server stub unpacks the parameters and calls the server.
6. The server does the work, returns the result to server stub.
7. The server stub packs it in a message and calls its local OS.
8. The server's OS sends the message to the client's OS.
9. The client's OS gives the message to the client stub.
10. The client stub unpacks the result and returns to the client.

Q7 (Problem 9.21). Suppose that two strangers A and B want to communicate with each other using secret-key cryptography, but do not share a key. Suppose both of them trust a third party C whose public key is well known. How can the two strangers establish a new shared secret key under these circumstances?

ANSWER: A and B pick random keys Ka and Kb and send them to C encrypted with C's public key. C picks a random key K and sends it to A encrypted using Ka and to B encrypted using Kb.

Q8 (Problem 9.26). Not having the computer echo the password is safer than having it echo an asterisk for each character typed, since the latter discloses the password length to anyone nearby who can see the screen. Assuming that passwords consist of upper and lower case letters and digits only, and that passwords must be a minimum of five characters and a maximum of eight characters, how much safer is not displaying anything?

ANSWER: It depends on how long the password is. The alphabet from which passwords is built has 62 symbols. The total search space is $62^5 + 62^6 + 62^7 + 62^8$, which is about 2×10^{14} . If the password is known to be k characters, the search space is reduced to only 62^k . The ratio of these is thus $2 \times 10^{14} / 62^k$. For k from 5 to 8, these values are 242,235, 3907, 63, and 1. In other words, learning that the password is only five characters reduces the search space by a factor of 242,235 because all the long passwords do not have to be tried. This is a big win. However, learning that it is eight characters does not help much because it means that all the short (easy) passwords can be skipped.

Q9 (Problem 9.8). Represent the ownerships and permissions shown in this UNIX directory listing as a protection matrix. *Note: asw* is a member of two groups: *users* and *devel*; *gmw* is a member only of *users*. Treat each of the two users and two groups as a domain, so that the matrix has four rows (one per domain) and four columns (one per file).

	Owner	*Group owner*	*Timestamp*	*File name*
-rw-r--r--	2 gmw	users 908	May 26 16:45	PPP-Notes
-rwx r-x r-x	1 asw	devel 432	May 13 12:35	prog1
-rw-rw----	1 asw	users 50094	May30 17:51	project.t
-rw-r-----	1 asw	devel 13124	May31 14:30	splash.gif

ANSWER: Here is the protection matrix:

Object				
Domain	PPP-Notes	prog1	project.t	splash.gif
asw	Read	Read Write Exec	Read Write	Read Write
gmw	Read Write	Read Exec	Read Write	
users	Read		Read Write	
devel		Read Exec		Read

Q10 (Problem 9.54). To verify that an applet has been signed by a trusted vendor, the applet vendor may include a certificate signed by a trusted third party that contains its public key. However, to read the certificate, the user needs the trusted third party's public key. This could be provided by a trusted fourth party, but then the user needs that public key. It appears that there is no way to bootstrap the verification system, yet existing browsers use it. How could it work?

ANSWER: Existing browsers come preloaded with the public keys of several trusted third parties such as the Verisign Corporation. Their business consists of verifying other companies' public keys and making up certificates for them. These certificates are signed by, for example, Verisign's private key. Since Verisign's public key is built into the browser, certificates signed with its private key can be verified.

THE END.