Problem 1. Input/Output

1. The speed of scanner: around 1 MB/s, WiFi 6: 9.6 Gbps, which is way faster. Therefore it is possible to transfer at a full speed.

2. A table named device table will have a record of the new device, and the entry of this table will contain a pointer to all the function of that device. What is more, entries can be updated, added, and removed. When a new device is added and installed on the operating system, a new entry is made in the table.

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

4. Since each bus transaction has a request and a response both taking 100 nsec, a bus transaction takes 200 nsec in total. To avoid a bottleneck, the bus has to be faster than the speed of the controller, which is

$$\frac{32\text{bit}}{200\text{ns}} = 20\text{MB/s}$$

5. A thin client is a computing dummy terminal in a client-server network that is essentially application-free. It communicates with the server via a number of protocols to access the local area network. The thin client sends its mouse, keyboard, and other inputs to the server for processing, and the server sends the results back to the thin client for display.

6. A mechanical hard drive is a conventional hard drive that consists of: a platter, a magnetic head, a platter spindle and control motor, a magnetic head controller, a data converter, an interface, a cache, and several other components.

   The magnetic head can be moved along the radius of the platter, and with the high speed of the platter spinning at thousands of revolutions per minute, the magnetic head can be positioned on the platter to read and write data. Information is written to the disk by an electromagnetic current that changes polarity through the magnetic head, which is very close to the magnetic surface, and information can be read in the opposite way. As a precision device, dust is the enemy of the hard disk, so the air entering the hard disk must be filtered.

7. RAID (Redundant Array of Independent Disks), formerly known as Redundant Array of Inexpensive Disks, is an abbreviation for disk array. Virtualized storage technology combines multiple hard drives into one or more disk arrays for the purpose of increasing performance or data redundancy, or both. In operation, depending on the RAID level, data is spread across the disks in a variety of ways, and the names of RAID levels begin with RAID numbers, such as: RAID 0, RAID 1, RAID 5, RAID 6, RAID 7, RAID 01, RAID 10, RAID 50, RAID 60. Each level has its own theoretical advantages

and disadvantages, and the different levels balance the two goals of increasing data reliability and increasing memory (group) read/write performance.

(a) RAID 0 is also known as striped set. It connects two or more disks in parallel to form a single high-capacity disk. When storing data, it is segmented and scattered among these disks, and since reads and writes can be processed in parallel, RAID 0 is the fastest of all levels. However, RAID 0 has neither redundancy nor fault tolerance, so if one disk (physically) fails, all data will be lost, which is as dangerous as JBOD.

(b) RAID 1 read speed is a multiple of the number of disks and is the same as RAID 0. There is a slight reduction in write speed. As long as one disk is functioning properly, it can maintain operation with the highest reliability. The principle is to store data on the primary hard disk and write the same data on the mirrored disk at the same time. When the primary (physical) hard disk fails, the mirrored hard disk takes over the work of the primary disk. Because mirrored hard disks are used for data backup, RAID 1 has the best data security of all RAID levels.

(c) RAID 5 is a storage solution that balances storage performance, data security, and storage cost. It uses Disk Striping technology. Instead of backing up the stored data, RAID 5 stores the data and corresponding parity information on each of the disks that make up RAID 5, and the parity information and corresponding data are stored on separate disks. RAID 5 can be thought of as a compromise between RAID 0 and RAID 1.

(d) RAID 6 adds a second, independent block of parity information compared to RAID 5. The two independent parity systems use different algorithms, and data reliability is very high, so data integrity will not be affected if any two disks fail simultaneously. "Depends strongly on the specific implementation, so RAID 6 is usually not implemented in software, but more likely in hardware.

(e) RAID 10 is a combination of partitioning the data and then mirroring it, and then dividing all the disks into two groups as if they were RAID 1 as the lowest combination, and then treating each group of RAID 1 as one "drive" operating as RAID 0.

Problem 2. Multiprocessors

1. We need to use the registers preloads one of its own registers with a 1, and then uses SWP to exchange the memory and the contents of the register. If it was locked before, then it is busy and waiting; otherwise it is locked now.

2. Virtualization is a resource management technique that takes the physical resources of a computer, abstracts them, transforms them into something that can be partitioned and combined into one or more computer configuration environments. This breaks down the inseparable barriers between physical structures and allows users to use these computer hardware resources in a better way than the original configuration.

   Main challenges:

(a) Whether the stability aspect can maintain the same stability as the physical machine and meet the extreme pursuit of response delay for the business.

(b) How virtualization technology can circumvent the impact of physical server downtime on virtual machines in the cloud, making it less or even less perceptible at the business level.

(c) The performance of virtualized computing, networking, storage, etc. is comparable to that of a physical machine, and whether it matches the performance of a physical machine.

Problem 3. File systems

1. `/usr/lib/mutt`

2. # of blocks in indirect block = 1kB / 4 b = 1024 / 4 = 256 blocks.
   Single direct block take 256 * 1 kB = 256 KB.
   Double indirect block will take 256 * 256 = 216 kB = 64 MB.
   Triple direct block will take 256 * 64 MB = 214 MB = 16 GB.
   In total, there are $13 + 256 + 64 * 1024 + 16 * 1024 * 1024 = 16843021KB$

3. For all block sizes, the rotation latency is

$$\frac{1000ms/s \times 60s/min}{15000rpm} \div 2 = 2s$$

   (a) $8 + 2 + \frac{1024}{262144} = 10.003ms$

   (b) $8 + 2 + \frac{2048}{262144} = 10.008ms$

   (c) $8 + 2 + \frac{4096}{262144} = 10.016ms$

Problem 4. Security

1. $26^8 = 208827064576$ with duplicated letters or $A_{26}^8 = 62990928000$.

   In cryptography, the size of the number determines the security it has. In the case of RSA, the current factoring record is the 829-bit RSA-250 in late Feb. 2020. Thus we must need have at least 1024 bit key length to ensure the security in the near future, and neither above result satisfies this condition.

2. `malloc` is only for allocating memory, not initializing the memory. Therefore, when the computer first boots up, the OS configures the memory controller and processor caches and initializes some devices, which usually set to 0. When some memory has been used by other program, the content will not be reset before provided to another new process.

3. Determined by the lowest level of security.

4. No, or at least questionable. The real weak link is humans, not the bugs in Microsoft's software."(McCue) The future seems to be full of more and more computers and with every increase in security comes an increase in the ability of people to take advantage of this ever growing number of computers. Even if security in computer systems could be secured with better programs, patches, anti-virus and IDS systems there will always be a huge security gap.....people. Now and in the future the weak-link in computer security is and will be users. No matter how secure a computer is, it cannot stop a person from sharing a password or intentionally compromising a system. The number of security events increases with the increase in the number of users and systems. [1]

Problem 5. Research

A firewall is an information security system that is placed between the Internet and an intranet to monitor incoming and outgoing traffic according to a predetermined corporate policy. A firewall may be a proprietary network device or may run on a host computer that checks network traffic across various network interfaces. It is one of the most important types of network protection devices available today, and professionally speaking, a firewall is a collection of hardware or software components that sit between two networks to provide inter-network access or control.

iptables

iptables[2], which uses tables to organize rule chains. Originated from freeBSD's ipfirewall (kernel 1.x era), all rules are in the kernel; renamed ipchains after 2.0x, multiple rules can be defined and used in a chain. iptablses slices and dices each of the five chains into five different tables according to its purpose and usage. This means that each table can be configured with separate rules for certain chains as needed. For example, rules can be configured for INPUT chains in both the mangle and filter tables, and the INPUT chain rules in both tables will be used for filter checking when a packet flows through the INPUT location (into user space).

iptables also supports custom rule chains. A custom chain must be associated with a particular chain. Rules can be set in a chain, and packets meeting certain conditions can be jumped to a destination chain for processing, and the destination chain can return to the current chain to continue processing subsequent rules. Since rules in a chain are checked sequentially from beginning to end, the order of the rules is very important. The stricter the rule, the higher it should be.

ebtables

ebtables is the Ethernet bridge firewall[3]. The Ethernet bridge works at the data-link layer, and ebtables filters data-link-layer packets. The 2.6 kernel has built-in ebtables. To use it, you must first install the ebtables user-space utility (ebtables-v2.0.6), after which you can

[1] Global Information Assurance Certification Paper
[2] http://drops.wooyun.org/tips/1424
[3] https://blog.csdn.net/u013485792/article/details/76522551

use ebtables to filter bridge packets. The rules for ebtables are as follows, depending on the user's requirements.

- For all packets, the default pass-through.
- Distinguish source and destination addresses and source and destination ports.
- Filter TCP, UDPP packets separately

ebtables is primarily used to control the data link layer. In the kernel, ebtables is more "up front" than iptables in terms of data intercept points, and the data it obtains is more "raw", ebtables is mostly used in bridge mode, such as controlling VLAN IDs, etc. ebtables is like iptables for an Ethernet bridge.

arptables

arptables is used to create, retrieve, and modify the kernel's arp packet processing tables. There are several different tables, each containing several built-in processing chains, and allowing the user to customize the processing chains[4].

Each chain is a list of rules, each of which matches a particular package. Each rule specifies an operation on the matching packet. This operation is also called a 'target', and the target can also be jumped to another chain in the same table.

Problem 6. Linux

A Kernel Panic is an action taken by an operating system when it detects an internal fatal error and cannot safely handle it. There are several reasons for this panic happen on my Mac OS Big Sur, such as high temperature, dual-screen output when connected with 32'4k monitor or iPad Pro. To have a detailed look into the kernel panic, we can visit the diagnostic report under`/lib/logs`, the following report logs one kernal panic.

```
1 Thread 0 Crashed:: Dispatch queue: com.apple.main-thread
2 0   libsystem_kernel.dylib               0x00007fff202f9462 __pthread_kill
  ↪ + 10
3 1   libsystem_pthread.dylib              0x00007fff20327610 pthread_kill +
  ↪ 263
4 2   libsystem_c.dylib                    0x00007fff2027a720 abort + 120
5 3   libsystem_malloc.dylib               0x00007fff2015b430 malloc_vreport
  ↪ + 548
6 4   libsystem_malloc.dylib               0x00007fff2016f702
  ↪ malloc_zone_error + 183
7 5   libsystem_malloc.dylib               0x00007fff20153182
  ↪ tiny_free_list_add_ptr + 1224
8 6   libsystem_malloc.dylib               0x00007fff2015045d
  ↪ tiny_malloc_from_free_list + 1640
```

[4]https://linux.die.net/man/8/arptables

```
7   libsystem_malloc.dylib                0x00007fff2014f876
    ↪ tiny_malloc_should_clear + 233
8   libsystem_malloc.dylib                0x00007fff2014e793
    ↪ szone_malloc_should_clear + 66
9   libsystem_malloc.dylib                0x00007fff20167dfe
    ↪ _malloc_zone_malloc + 118
10  libc++abi.dylib                       0x00007fff202eddaa operator
    ↪ new(unsigned long) + 26
11  libstdc++.6.dylib                     0x00007fff78acf2e7
    ↪ std::string::_Rep::_S_create(unsigned long, unsigned long,
    ↪ std::allocator<char> const&) + 89
12  libstdc++.6.dylib                     0x00007fff78acf579
    ↪ std::string::_M_mutate(unsigned long, unsigned long, unsigned long) + 87
13  libstdc++.6.dylib                     0x00007fff78ad01b8
    ↪ std::string::assign(char const*, unsigned long) + 78
14  UpdaterStartupUtility                 0x000000010efe4e1f 0x10efc9000 +
    ↪ 114207
15  UpdaterStartupUtility                 0x000000010efe46f1 0x10efc9000 +
    ↪ 112369
16  UpdaterStartupUtility                 0x000000010efccd36 0x10efc9000 +
    ↪ 15670
17  UpdaterStartupUtility                 0x000000010efccb1f 0x10efc9000 +
    ↪ 15135
18  UpdaterStartupUtility                 0x000000010efcab39 0x10efc9000 +
    ↪ 6969
19  libdyld.dylib                         0x00007fff20342631 start + 1

Thread 1:
0   libsystem_pthread.dylib               0x00007fff20323458 start_wqthread
    ↪ + 0

Thread 2:
0   libsystem_pthread.dylib               0x00007fff20323458 start_wqthread
    ↪ + 0

Thread 0 crashed with X86 Thread State (64-bit):
  rax: 0x0000000000000000  rbx: 0x0000000117c7ae00  rcx: 0x00007ffee0c35e78
    ↪  rdx: 0x0000000000000000
  rdi: 0x0000000000000307  rsi: 0x0000000000000006  rbp: 0x00007ffee0c35ea0
    ↪  rsp: 0x00007ffee0c35e78
   r8: 0x0000000000000000   r9: 0x0000000000000000  r10: 0x0000000117c7ae00
    ↪  r11: 0x0000000000000246
  r12: 0x0000000000000307  r13: 0x0000000000000043  r14: 0x0000000000000006
    ↪  r15: 0x0000000000000016
  rip: 0x00007fff202f9462  rfl: 0x0000000000000246  cr2: 0x000000010f204000
```

To trouble shoot, I can start with hardwares or softwares. I can disconnect all the components and devices from my computer. Another approach is to kill all the process and hard-reboot the computer to avoid the panic brought by maximum usage of ram or conflict between two softwares.

Problem 7. Course survey

Completed! :)