VE482 Homework 8

Liu Yihao 515370910207

Ex. 1 — Input/Output

1. Yes, a typical scanner runs within 1 MB/s, and a wireless network nowadays usually exceeds this speed. So it can be sent in full speed.

2. The OS can save the driver's interface in a table, when a new device is installed, it only need to update the table so that the new device can be used.

3. (a) Device driver

   (b) Device driver

   (c) Device-independent software

   (d) User-level software

4. $$\frac{32\,\text{bit}}{100\,\text{ns}\times 2} = 2\times 10^{7}\,\text{B/s} = 20\,\text{MB/s}$$

5. A thin client is a lightweight computer built to connect to a server from a remote location. The server does most of the work, which can include crunching numbers and storing information for the thin client. In contrast, a conventional desktop PC (fat client) does most of the work but can communicate with and run programs on a server.

   Reference: https://en.wikipedia.org/wiki/Thin_client

6. A modern HDD records data by magnetizing a thin film of ferromagnetic material on a disk. Sequential changes in the direction of magnetization represent binary data bits. The data is read from the disk by detecting the transitions in magnetization. User data is encoded using an encoding scheme, such as run-length limited encoding, which determines how the data is represented by the magnetic transitions.

   A typical HDD design consists of a spindle that holds flat circular disks, also called platters, which hold the recorded data. The platters are made from a non-magnetic material, usually aluminum alloy, glass, or ceramic, and are coated with a shallow layer of magnetic material typically 10–20 nm in depth, with an outer layer of carbon for protection. For reference, a standard piece of copy paper is 0.07–0.18 millimeters (70,000–180,000 nm).

   Reference: https://en.wikipedia.org/wiki/Hard_disk_drive

7. - RAID 0 consists of striping, without mirroring or parity. The capacity of a RAID 0 volume is the sum of the capacities of the disks in the set, the same as with a spanned volume. There is no added redundancy for handling disk failures, just as with a spanned volume.

- RAID 1 consists of data mirroring, without parity or striping. Data is written identically to two drives, thereby producing a ”mirrored set” of drives. Thus, any read request can be serviced by any drive in the set. If a request is broadcast to every drive in the set, it can be serviced by the drive that accesses the data first (depending on its seek time and rotational latency), improving performance.
- RAID 5 consists of block-level striping with distributed parity. Unlike RAID 4, parity information is distributed among the drives, requiring all drives but one to be present to operate. Upon failure of a single drive, subsequent reads can be calculated from the distributed parity such that no data is lost. RAID 5 requires at least three disks. RAID 5 implementations are susceptible to system failures because of trends regarding array rebuild time and the chance of drive failure during rebuild.
- RAID 6 consists of block-level striping with double distributed parity. Double parity provides fault tolerance up to two failed drives. This makes larger RAID groups more practical, especially for high-availability systems, as large-capacity drives take longer to restore. RAID 6 requires a minimum of four disks. As with RAID 5, a single drive failure results in reduced performance of the entire array until the failed drive has been replaced.
- RAID 10, also called RAID 1+0 and sometimes RAID 1&0, is similar to RAID 01 with an exception that two used standard RAID levels are layered in the opposite order; thus, RAID 10 is a stripe of mirrors.

Ex. 2 — Multiprocessors

1. We can preload 1 into the register to be used, and after the SWP, the memory is 1, so it is locked. If it was previously locked, the register is still 1, then the function should loop until it becomes 0.

2. Virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.

Ex. 3 — File systems

1. `/usr/lib/mutt`

2. $$\frac{1\,\text{KB}}{4\,\text{B}} = 256$$
   $$13 + 256 + 256^2 + 256^3 = 16843021\,\text{KB}$$

3. (i)
   $$8 + 2 + \frac{1024}{262144} = 10.003\,\text{ms}$$
   (ii)
   $$8 + 2 + \frac{2048}{262144} = 10.008\,\text{ms}$$
   (iii)
   $$8 + 2 + \frac{4096}{262144} = 10.016\,\text{ms}$$

Ex. 4 — Security

1. $$C_{26}^{8} = 1562275$$

   There are 1562275 possible words. It's easy to apply a bruce-force attack on such a small range of password, so a password should always contain lower and uppercase letters, numbers and symbols.

2. When the computer is powered on, all of the memory is 0. However, when a process ended, the memory it had used is not necessary to be cleaned. So the allocated memory is sometimes random and some other times set to 0.

3. The lowest level of security among all components.

4. No. The system is always interfered by outer issues and never can be guaranteed to be secure.

Ex. 5 — Research

iptables

iptables is a user-space utility program that allows a system administrator to configure the tables provided by the Linux kernel firewall (implemented as different Netfilter modules) and the chains and rules it stores. Different kernel modules and programs are currently used for different protocols; iptables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebtables to Ethernet frames.

iptables requires elevated privileges to operate and must be executed by user root, otherwise it fails to function. On most Linux systems, iptables is installed as /usr/sbin/iptables and documented in its man pages, which can be opened using man iptables when installed. It may also be found in /sbin/iptables, but since iptables is more like a service rather than an "essential binary", the preferred location remains /usr/sbin.

The term iptables is also commonly used to inclusively refer to the kernel-level components. x_tables is the name of the kernel module carrying the shared code portion used by all four modules that also provides the API used for extensions; subsequently, Xtables is more or less used to refer to the entire firewall (v4, v6, arp, and eb) architecture.

Reference https://en.wikipedia.org/wiki/Iptables

ebtables

ebtables is an application program used to set up and maintain the tables of rules (inside the Linux kernel) that inspect Ethernet frames. It is analogous to the iptables application, but less complicated, due to the fact that the Ethernet protocol is much simpler than the IP protocol.

There are three ebtables tables with built-in chains in the Linux kernel. These tables are used to divide functionality into different sets of rules. Each set of rules is called a chain. Each chain is an ordered list of rules that can match Ethernet frames. If a rule matches an Ethernet frame, then a processing specification tells what to do with that matching frame. User-defined chains are very useful to get better performance over the linear traversal of the rules and are also essential for structuring the filtering rules into well-organized and maintainable sets of rules.

Reference: h3.0.ttp://ebtables.netfilter.org/misc/ebtables-man.html

arptables

arptables is a user space tool, it is used to set up and maintain the tables of ARP rules in the Linux kernel. These rules inspect the ARP frames which they see. arptables is analogous to the iptables user space tool, but arptables is less complicated.

The kernel table is used to divide functionality into different sets of rules. Each set of rules is called a chain. Each chain is an ordered list of rules that can match ARP frames. If a rule matches an ARP frame, then a processing specification tells what to do with that matching frame. The processing specification is called a ‘target’. However, if the frame does not match the current rule in the chain, then the next rule in the chain is examined and so forth. The user can create new (user-defined) chains which can be used as the ‘target’ of a rule.

Reference: http://ebtables.netfilter.org/misc/arptables-man.html