

2019 Operating System Examination Solution

1.
 - a. The purpose of Bus 1 is to provide communication between CPU and memory and everything else, such as other devices and chips. Bus 2 is used to provide communication between peripherals (e.g. graphics card) and CPU. It might be PCIe. Bus 3 is used for accessing data from hard disk. There main difference is the bandwidth and speed: Bus 1 has the highest bandwidth and lowest latency as communications from/to CPU need to be fast. Bus 3 is the slowest and has lowest bandwidth as the speed of hard disk cannot match with the speed of main memory/CPU.
 - b. The OS has to ensure fair access to shared devices, ensure fast responses from/to I/O, need to enable parallel process between I/O and processes, and provide simple and uniform interface to hide the ugly implementation of device handling.
 - c. Character devices transfer data byte to byte without buffering, while the block devices use an intermediate buffer to cache data. Character devices include mouse, keyboard, monitor, etc. Block devices mainly include hard disks and any other long term storage devices.
 - d. Character device, such as graphic card, sound card, USB, might be wired in bus 2 in order to communicate with CPU, while the block devices, such as hard disk and other storage devices, might be wired in bus 3.
 - e.
 - i) NVRAM would connect to Bus 1 since it offers the access bandwidth of memory, which needs the bus with matching bandwidth
 - ii) NVRAM would likely to be a block device since it has persistent storage, which may be accessed with a system buffer.
 - iii) The abstraction needs to include `read()`, `write()`, `seek()`, `ioctl()`, etc, in order to visit and control NVRAM. The manufacturer of NVRAM (e.g. the person who writes the device driver for NVRAM) needs to ensure they implement those functions in order to enable user space program to access NVRAM.
2.
 - a. It might store file size, creation date, last modified date, file type (e.g. regular or device), access control information, etc.
 - b. The pointers can be direct pointers, indirect pointer, double indirect pointer, or triple indirect pointer. Because file size might vary in a large range, we need to ensure that we can easily store small files using direct pointers while still be able to store large files using double/triple indirect pointers.
 - c. For `wombat.txt`, it is sufficient to use 3 direct pointers and 1 indirect pointer (i.e. pointer 1-3 are direct and pointer 4 is indirect). Since the block size is 4096 bytes = 4 KBytes, 3 pointers are able to store the first $3 \cdot 4 = 12$ KBytes of the file, while the rest 4 KBytes are stored using the indirect pointer.

For `dingo.txt`, it needs the first three direct pointer, the indirect pointer, and the double indirect pointer. The first 3 pointers will store $3 \cdot 4 = 12$ KBytes, and the indirect pointer will store $\frac{4096}{4} \cdot 4096 = 1024 \cdot 4$ KBytes = 4096 KBytes. This adds up to $12 + 4096 = 5008$ KBytes stored and $8209 - 5008 = 3201$ KBytes left to be stored. It is sufficient to use a double indirect pointer to store the rest.

d. (Diagram omitted)

e. Storing a large number of files with an average size of 10 bytes will result in large internal fragmentation caused by the unused space within each single block. We allow one single block to store multiple different files of small sizes to reduce such internal fragmentation. We need to store an extra field to record the offset within the block to indicate the starting point of that file in the block. This offset is only used for the first pointer as files with an average size of 10 bytes will not use pointers beyond the first one.