Solution to HW1

1.6:

A program may be delayed or aborted if a real device is not available to meet its request and a virtual device is not created. Hence, creation of virtual devices improves user service. It may increase the number of programs the OS can service simultaneously, so it may also improve efficiency of use.

2.2:

The ‘mode (M)’ field of the PSW indicates whether the CPU is in the kernel mode. The kernel forms an ‘initial PSW’ for a program when the program is to be initiated. The PC field of this PSW points to the first instruction of the program, and its M field is set to 1 to indicate that the CPU is in the user mode. Let a program \textit{prog1} be in execution when an interrupt occurs. Contents of the PSW are saved in some kernel data structure corresponding to \textit{prog1} (later in Chapter 3 we call it the PCB). A new PSW is loaded into the CPU by the interrupt action. The M field of this PSW is 0, so the CPU is in the kernel mode. The saved PSW of \textit{prog1} is loaded back into the CPU when \textit{prog1} is to be resumed. This way, the CPU enters the kernel mode when the kernel gets control for execution, and the CPU enters the user mode when a user program is in execution (See Example 2.3).

2.8:

When a software interrupt signifying a system call occurs, the interrupt hardware transfers control to the kernel. After processing the interrupt, the kernel passes control to a user program. Both these actions cause switching between programs. This overhead can be reduced by making a single request to obtain a large chunk of memory, instead of making several requests for small areas of memory.

2.9:

At a cache miss in the higher cache level, a cache block would be transferred from the lower cache level. Hence the effective access time is governed by the following equation, where \textit{h cache} and \textit{l cache} denote the higher and
lower cache levels, respectively, and $h$-tra and $l$-tra denote transfer of a block to the higher and lower cache levels, respectively.

$$t_{ema} = h \cdot t_{h\text{-cache}} + (1 - h) \cdot ((h \cdot t_{h\text{-tra}} + (1 - h) \cdot (t_{l\text{-tra}} + t_{h\text{-tra}})) + t_{h\text{-cache}})$$

$$= h \cdot t_{h\text{-cache}} + (1 - h) \cdot (t_{h\text{-tra}} + (1 - h) \cdot t_{l\text{-tra}} + t_{h\text{-cache}})$$

Since the access time of memory is 10 microseconds, $t_{l\text{-cache}}$ and $t_{h\text{-cache}}$ are 1 microsecond and 0.1 microsecond, respectively. Accordingly,

$$t_{ema} = 0.9 \times 0.1 + 0.1 \times ((5 \times 1 + 0.1 \times 5 \times 10) + 0.1)$$

which is 1.10 microseconds