VE482 Homework 1

Liu Yihao 515370910207

Ex. 1 — Revisions

Stack is a region of memory reserved by each thread, it is faster than a heap, because the data is added and removed in a last-in-first-out manner. When a function is called, all of the data it needs are saved in the stack one by one. When the function exits, these data can be easily removed. The length and position of a variable in a function is defined by the compiler, which can't be dynamic. So stack-based allocation is suitable for temporary data in function calls.

Heap is a region of memory managed by operating systems, it is typically larger than a stack, and the precise location of the allocation is not known in advance, the memory is accessed indirectly, usually through a pointer reference. A program can apply and release memory through system calls during executing. When a process ends, the operating system will release all of the heap memory it used.

Ex. 2 — Personal Research

1. When the computer is powered on, the CPU executed a program contained in ROM (like BIOS) at a predefined address. The program (BIOS as example) searches for devices installed on the computer and loads them and tries a power-on self-test. If no error occurs, it locate boot loader software held on a storage device. It loads and executes the first boot software (often an OS) it finds, giving it control of the PC. The role of BIOS is a bridge between the PC and the operating system.

2. Hybrid kernels combine micro kernels and monolithic kernels.
   Exo kernels are small and give more direct access to the hardware.

Ex. 3 — Course application

1. a) Disable all interrupts should only be allowed in kernel mode because the process of interrupt is in kernel mode.

   b) Read the time-of-day clock should only be allowed in kernel mode because it needs I/O with the hardware (CMOS).

   c) Set the time-of-day clock should only be allowed in kernel mode because it needs I/O with the hardware (CMOS).

   d) Change the memory map should only be allowed in kernel mode because memory management is in kernel mode.

2. If the OS is smart enough, since there are four logic cores in the system, three programs can be executed in parallel, 20 ms is needed. However, if the OS is not so smart, some or all of the programs can't be executed in parallel, then maybe 25 ms, 30 ms, 35 ms is needed.

Ex. 4 — Simple problem

The RAM needed to support a 25 lines by 80 rows character monochrome text screen is $25 \times 80/8 = 250\,\mathrm{B}$. It costs $250 \times 5/1024 = \$1.22$.

The RAM needed to support a 1024 x 768 pixel 24-bit color bitmap is $1024 \times 768 \times 24/8 = 2304\,\mathrm{KB}$. It costs $2304 \times 5 = \$11570$.

Now the cost if RAM is about $10/ GB.

Ex. 5 — Command lines on a Unix system

```bash
#bin/bash
# 1. Create a new user
useradd newuser
# 2. List all the currently running processes
ps -e
# 3. Display the characteristics of the CPU and the available memory
top
# 4. Redirect some random output into two different files
od /dev/urandom | head | tee 1.txt > 2.txt
# 5. Concatenate the two previous files
cat 1.txt 2.txt > 3.txt
# 6. Read the content of the resulting file as hexdecimal values (in other words find a command to read a file as hexadecimal values)
od -t x 3.txt
# 7. Use a single command to find all the files in /usr/src with the word semaphore in their name and containing the word ddekit sem down
find /usr/src -name '*semaphore*' | xargs grep -lw 'ddekit_sem_down'
```

The shell file is named “ex5/ex5.sh”.