UM-SJTU JOINT INSTITUTE
VE482 Introduction to Operating Systems

Homework 1

Li Yong 517370910222

October 3, 2020

Ex.1 Revisions

stack: the space which is allocated automatically by the compiler. Hence it takes less time to allocate the space. It stores the parameters of functions, local variables and etc. The address is decreasing and *stack* consists of contiguous memory space, i.e., the space of *stack* is limited. If the space we need is more than the left space of *stack*, then overflow occurs. The operations of *stack* are similar to data structure stack.

heap: the space which is allocated by programmers, generally. Programmer also need to declare the size of the space and free it after the usage. So that it takes more time to allocate the space. It stores the space allocated by programmers. The address is increasing and *heap* consists of discontinuous space. The size of *heap* is determined by the size of virtual memory. The operations of *heap* are similar to linked list, because OS stores the addresses of spare space in linked list.

Ex.2 Personal Research

1. When a computer is powered on, a signal will be passed to south bridge to reset. Then north bridge and CPU will reset. Then POST, Power On Self Test, which intends to check the hardwares, was executed by BIOS. If there is no hardwares issues, BIOS will call *boot loader* on boot disk. Boot loader's job is to start the real operating system by looking for a kernel. Once the kernel starts, the loading of operation system begins.

2. **Hybird kernels:** Hybrid kernel is a kernel architecture based on a combination of microkernel and monolithic kernel architecture used in computer operating systems.
   Exo kernels: The exokernel architecture is designed to separate resource protection from management to facilitate application-specific customization.

Ex3. Course application

1. a)c)d) In the user mode, it can only execute a subset of all instructions. These options are critical instructions, so that user mode does not have the authority.

2. Let's mark the CPUs as CPU1 and CPU2.

   (a) P0: CPU1 P1: CPU1 P2: CPU2 20ms

   (b) P0: CPU1 P1: CPU2 P2: CPU1 25ms

   (c) P0: CPU1 P1: CPU2 P2: CPU2 30ms

   (d) P0: CPU1 P1: CPU1 P2: CPU1 35ms

Ex.4 Simple problem

As for a character monochrome text screem, each pixel is 1 byte.

$$80 \times 25 = 2000\ bytes = 2KB,$$

which costs $10.

As for a 1024× 768 pixel 24-bit color bitmap,

$$1024 \times 768 \times 24 = 2359.296KB,$$

which costs $11,796.48. Now it is about $0.95/MB.

Ex.5 Command lines on a Unix system

```
#!/bin/Ash
#VE482 Homework1
useradd -d /home/wenator04 wenator04 #Create a new user
ps -ax #List all the currently running processes
top #Display the characteristics of the CPU and the available memory
head -200 /dev/urandom | cksum | cut -f 1 -d"␣" > random1
head -200 /dev/urandom | cksum | cut -f 1 -d"␣" > random2 #Redirect some random output into two different files
cat random1 random2>concatenate #Concatenate the two previous files
hexdump concatenate #Read the content of the resulting file as hexdecimal values
find /usr/src -type f -name "*semaphore*" | xargs grep -rl "ddekit_sem_down" #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
```

.sh file is also attached.