

# Week4-Assignment-report

---

11911609-葛兆宁

## Q1

- (1) 关于三个重要的pieces, 在阅读完第1章以及第2章之后, 我的理解如下: 首先是虚拟化, 虚拟化即将电脑的物理资源 (如cpu资源, 内存资源等) 抽象为人们逻辑所能理解且容易理解的内容, 通过用户的逻辑输入能够自动或非自动的调整对资源的管理。其次是并发, 即多个process同时执行, 这通常靠cpu多线程并行。最后是永久保存的管理, 操作系统还要妥善管理那些硬件存储, 以保证计算机能正常使用这些资源 (包括文件系统, IO等等)。
- (2) 关于操作系统龙书 (我暂且这么称呼, 以区别编译原理龙书), 他的3、4、5、6、7、8、9章可以属于并发内容以及并发会出现的问题, 3、4、9、10、18、19属于虚拟化内容, 11、12、13、14、15章对应的是永久保存系统, 至于16、17章属于安全问题, 我暂且没想好怎么分类。

## Q2

- 首先, OS会保存当前进程的寄存器, PC与kernelStack中的当前进程的位置指针, 然后将PC与寄存器加载下一个将执行进程的东西, 内存指针会进行一个switch call的操作, 将指针指向下一个进程的kernelStack的地址, 最后OS会执行return-from-trap的操作, 接着完成上下文切换, 下一个进程开始跑动

## Q3

- (1) 关于fork()函数的system-call调用, 它的机制是从操作系统生成一个和父进程相同内容的子进程, 然后通过上下文切换同时进行父子进程。具体过程为首先是PCB的复制, 然后是改变子进程的PCB, 设置新的pid, 将运行时间归零, 同时更新父进程指针, 父进程则更新子进程数组, 将新的子进程加入到子进程数组中。之后, 再在内存空间中为子进程复制一份新的地址空间, 将子进程的PCB中的地址空间指针指向新的内存空间, 最后将子进程的PCB放入TaskList一起执行
- (2) 当OS执行exit()函数时, 进程结束, 资源全部回收, 只在taskList里挂一个空的PCB, 同时发出信号。这时, 如果父进程结束运行了, 进程归init进程管, 则tasklist中去掉PCB; 若父进程未结束且处于wait()状态, 则收到信号后在tasklist中删除PCB; 若父进程未结束运行, 且不处于wait状态, 则子进程进入僵尸态

## Q4

- 三种方式分别为主动, 被动与自动 (我自己总结的)。其中主动为用户主动直接或间接调用system-call, 操作系统会根据system-call的要求转换到内核态。被动是指操作系统在执行用户程序时遇到中断, 这时操作系统会转移至内核态去执行相应的操作, 最后为自动的, 即在用户的执行完某些操作后, 操作系统会自动切换至内核态执行后续或伴随的一些操作

## Q5

- 一个life-cycle包含了Ready, Blocked, Running, Zombie/Terminated四个状态。其中, Ready一般为刚被创建或Blocked状态达到预期的响应后, 进程转入Ready状态。Ready状态通过OS的调度会选择

性的进入Running状态，同样Running也会通过OS调度进入Ready状态。Running状态会通过wait()或中断等操作Blocked状态，然后进程执行完后会从Running状态进入Zombie/Terminated状态。

## Q6

- shell.c:

```
#include <unistd.h>
# include <stdio.h>
# include <string.h>
# include <strings.h>
# include <sys/wait.h>
# include <malloc.h>
int main()
{
    // initialise present user directory
    char address[200] = "~/OS/assignments/week4";
    while (1)
    {
        //get present user directory
        getcwd(address, 200 *sizeof(char));

        //print out the user account & user directory
        printf("11911609JohnnyGe@johnny-Ge-WXX9:%s$ ", address);

        //command arguments initialise
        char*args[5];

        // number of the arguments, default=0
        int length = 0;

        // scan in unit
        char ch;
        int j = 0;

        // use to temporarily store the argument
        char arg[10];

        // input command
        while (ch = getchar())
        {
            // the all command end
            if (ch == '\n')
            {
                arg[j] = '\0';
                char *argl = (char*)malloc((j + 2) *sizeof(char));
                strcpy(argl, arg);
                args[length] = argl;
                length++;
                j = 0;
                break;
            }
        }
    }
}
```

```
// one argument input end
if (ch == ' ')
{
    arg[j] = '\0';
    char*argl = (char *)malloc((j + 2)* sizeof(char));
    strcpy(argl, arg);
    args[length] = argl;
    length++;
    j = 0;
}

// combine to a complete argument
else
{
    arg[j++] = ch;
}
}

// initialise the space that has not argument
for (int i = length; i < 5; i++)
{
    args[i] = NULL;
}

// get command and execute
if (length != 0)
{
    // execute cd
    if (strcmp("cd", args[0]) == 0)
    {
        if (chdir(args[1]) != 0)
        {
            printf("No such directory\n!");
        }
    }

    // execute command in /bin directory
    else
    {
        if (fork())
        {
            wait(NULL);
            // return the argument space to OS
            for (int i = 0; i < 5; i++)
            {
                free(args[i]);
            }
        }

        // execute command in child process
        else
        {
            char filename[20] = "/bin/";
```

```

        strcat(filename, args[0]);
        execl(filename, args[0], args[1], args[2],
args[3], args[4], NULL);
    }
}
}
}
}
}

```

- output:

```

11911609JohnnyGe@johnny-Ge-WXX9:~/OS/assignments/week4$ g++ myshell.c
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/assignments/week4$ ./a.out
11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments/week4$ ls
11911609-葛兆宁-week4-report.md  a.out  image  myshell.c  'Week_4_Assignment(in English).pdf'
11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments/week4$ ps
  PID TTY          TIME CMD
  5850 pts/0    00:00:00 bash
  6334 pts/0    00:00:00 a.out
  6439 pts/0    00:00:00 ps
11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments/week4$ pwd
/home/11911609JohnnyGe/OS/assignments/week4
11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments/week4$ ls -a
.  ..  11911609-葛兆宁-week4-report.md  a.out  image  myshell.c  'Week_4_Assignment(in English).pdf'
11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments/week4$ cd ..
11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments$ cd week
No such directory
!11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments$ cd sdk
No such directory
!11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments$ cd week4
11911609JohnnyGe@johnny-Ge-WXX9:/home/11911609JohnnyGe/OS/assignments/week4$ █

```