

操作系统作业一

要求：独立完成，严禁抄袭，按时提交。完成作业前请先阅读教材对应章节，回答问题须逻辑清晰、语言简练、准确全面。

提交截至日期：3.31（周日）18:00。

提交方式：提交 PDF 版本，PDF 命名规则为学号_姓名_作业 1，比如 PB220110000_张三_作业 1.pdf。

1. 阐述操作系统的功能，详细列举操作系统提供的服务。
2. 什么是 storage hierarchy？阐述缓存的思想与工作原理。
3. 什么是系统调用？系统调用的参数如何传递？阐述系统调用与库函数的区别，以及系统调用与 API 的逻辑关系。
4. 什么是 Dual Mode 机制？阐述其工作原理，以及采用 Dual Mode 的原因。
5. 分别阐述宏内核结构，层次化结构，模块化结构和微内核结构的特点和优劣。
6. 举例解释机制与策略分离的设计原则，阐述该设计的好处。

7. 除了初始父进程,分析以下程序共创建了多少子进程?

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    int i;

    for (i = 0; i < 4; i++)
        fork();

    return 0;
}
```

8. 分析以下程序，给出A, B, C, 和D位置的 pid 的取值(假设父子进程的pid分别是600和603)。

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid, pid1;

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        pid1 = getpid();
        printf("child: pid = %d",pid); /* A */
        printf("child: pid1 = %d",pid1); /* B */
    }
    else { /* parent process */
        pid1 = getpid();
        printf("parent: pid = %d",pid); /* C */
        printf("parent: pid1 = %d",pid1); /* D */
        wait(NULL);
    }

    return 0;
}
```

9. 分析以下程序, 阐述 LINE X 和 LINE Y 的输出.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

#define SIZE 5

int nums[SIZE] = {0,1,2,3,4};

int main()
{
    int i;
    pid_t pid;

    pid = fork();

    if (pid == 0) {
        for (i = 0; i < SIZE; i++) {
            nums[i] *= -i;
            printf("CHILD: %d ",nums[i]); /* LINE X */
        }
    }
    else if (pid > 0) {
        wait(NULL);
        for (i = 0; i < SIZE; i++)
            printf("PARENT: %d ",nums[i]); /* LINE Y */
    }

    return 0;
}
```

10.分析以下程序，阐述LINE Y是否会被执行，为什么？

```
int main(void) {  
    printf("before execl ...\n");  
    execl("/bin/ls", "/bin/ls", NULL);  
    printf("after execl ...\n");    /*LINE: Y*/  
    return 0;  
}
```