

# OS\_Assignment4

Assign	
Property	
tag	homework
姓名	
学号	

## 1.进程创建

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>

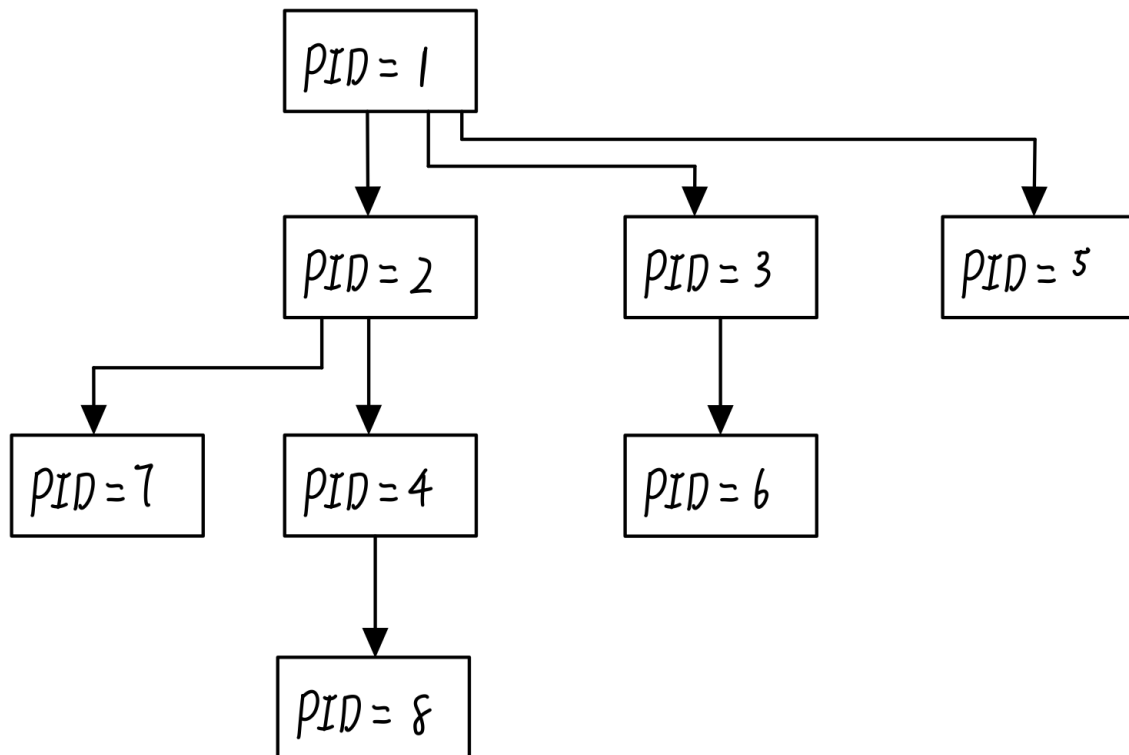
int main() {
    pid_t pid;
    int i;
    for (i = 0; i < 3; ++i) {
        pid = fork();// fork another process
        if (pid < 0) { // error occurred
            fprintf(stderr, "Fork Failed");
            exit(-1);
        } else if (pid == 0) {
            fprintf(stdout, "i=%d, pid=%d, parent pid=%d\n", i, getpid(),
getppid());
        }
        wait(NULL);
        exit(0);
    }
}
```

请问该程序最终一共生成几个进程?假设当前进程 PID 为 1, 生成的进程 PID 依次加 1, 请将生成进程关系图画出来。

运行结果为：

```
i=0, pid=17296, parent pid=17295
i=1, pid=17297, parent pid=17295
i=2, pid=17298, parent pid=17295
i=1, pid=17299, parent pid=17296
i=2, pid=17300, parent pid=17297
i=2, pid=17301, parent pid=17296
i=2, pid=17302, parent pid=17299
```

故共创建出7个线程, 其关系如下



## 2、程序题

程序题)写一个 2 线程的程序，首先生成一个从 1 到 1000 万的整数数组，然 后用两个线程分别计算数组奇数部分和偶数部分的和，并打印出总的和。分别 在单核和双核系统上运行该程序，计算加速比。(采用 pthread API) 提示:单核和 双核可在虚拟机上配置;pthread 调用方法网上有大量资料，比如 [https://www.ibm.com/developerworks/cn/linux/thread/posix\\_thread1/index.html](https://www.ibm.com/developerworks/cn/linux/thread/posix_thread1/index.html)

```

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define MAXLENGTH 10000000

//int pthread_create(pthread_t * tid, const pthread_attr_t * attr, void * ( * func) (void * ), void * arg)
//int pthread_join (pthread_t tid, void ** status);
int data[MAXLENGTH];
int cnt = 100;
void *sum(void *input);
unsigned long long res = 0;
int main() {
    while (cnt--){
        unsigned long long *odd, *even;
        pthread_t mythread;
        for (int i = 0; i < MAXLENGTH; ++i) {
            data[i] = i + 1;
        }
        //created thread
    }
}

```

```

        pthread_create(&mythread, NULL, sum, (void*)0);
        pthread_join(mythread, (void **)&odd);
        //current thread
        even = sum((void *)1);
        /*
        * pthread_create(&mythread, NULL, sum, (void*)1);
        * pthread_join(mythread, (void **)&even);
        *
        * */
        res = *even + *odd;
        free(odd);
        free(even);
    }
    printf("%llu\n", res);

    return 0;
}
void *sum(void *input){
    int i = (unsigned long long )input;
    unsigned long long *res = malloc(sizeof (unsigned int));
    for ( *res = 0; i < MAXLENGTH ; i = i + 2) {
        *res += data[i];
    }
    return res;
}

```

可以注意到有一部分代码被注释掉了，第一次写的时候创建了两个线程，但我没有意识到在程序正在运行时就已经是一个current thread了，换言之这相当于写了一个多线程程序，最后产生的结果是灾难性的，因为甚至出现了单核运行时间低于双核运行时间的情形。在作出适当修改后，结论如下：

双核对单核的加速比为：

$$\frac{T_m}{T_s} = \frac{3.744}{3.643} = 1.027$$

```

stu@stu:~/0Shomo$ time ./thread
50000005000000

real  0m3.744s
user  0m3.716s
sys  0m0.020s

```

```

stu@stu:~/0Shomo$ time ./thread
50000005000000

real  0m3.643s
user  0m3.625s
sys  0m0.024s

```

```

stu@stu:~/0Shomo$ time ./thread1
50000005000000

real  0m3.693s
user  0m3.682s
sys  0m0.004s

```

```
stu@stu:~/0Shomo$ time ./thread1
50000005000000

real  0m3.675s
user  0m3.651s
sys   0m0.016s
```