lab3-Linux环境下调试与矩阵乘法优化

朱宏基　　220110131　　大二(上)

实验内容:

`fork()` 的返回值中，子进程对应的是 `0` ， 父进程对应的是 `pid` 。

改造1:

在 `p1.c` 文件中针对父子进程分别调用sleep函数，改造代码如下：

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

int main(int argc, char *argv[])
{
    printf("hello world (pid:%d)\n", (int)getpid());
    int rc = fork();
    if (rc < 0)
    {
        // fork failed; exit
        fprintf(stderr, "fork failed\n");
        exit(1);
    }
    else if (rc == 0)
    {
        // child (new process)
        // sleep(5);          第一次改造让父进程休眠10s,子进程休眠5s
        sleep(15); // 第二次改造让父进程休眠10s,子进程休眠15s
        printf("hello, I am child (pid:%d)\n", (int)getpid());
    }
    else
    {
        // parent goes down this path (original process)
        sleep(10);
        printf("hello, I am parent of %d (pid:%d)\n",
               rc, (int)getpid());
    }
    return 0;
}
```

两次休眠结果如下：

```
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ gcc -o p1 p1.c
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ time ./p1
hello world (pid:14053)
hello, I am child (pid:14054)
hello, I am parent of 14054 (pid:14053)

real    0m10.001s
user    0m0.001s
sys     0m0.000s
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ gcc -o p1 p1.c
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ time ./p1
hello world (pid:14229)
hello, I am parent of 14230 (pid:14229)

real    0m10.001s
user    0m0.001s
sys     0m0.000s
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ hello, I am child (pid:14230)
```

发现 `time` 命令测量的 `real`

改造2:

在 `p1.c` 中首先导入头文件 `common.h` 然后同前面调用 `sleep` 函数一般改为调用 `Spin` 函数，改造代码如下：

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "common.h"

int main(int argc, char *argv[])
{
    printf("hello world (pid:%d)\n", (int)getpid());
    int rc = fork();
    if (rc < 0)
    {
        // fork failed; exit
        fprintf(stderr, "fork failed\n");
        exit(1);
    }
    else if (rc == 0)
    {
        // child (new process)
        // sleep(5); // 第一次改造让父进程休眠10s,子进程休眠5s
        // sleep(15); // 第二次改造让父进程休眠10s,子进程休眠15s

        // Spin(5); // 第一次改造让父进程Spin10s,子进程Spin5s
        Spin(15); // 第二次改造让父进程Spin10s,子进程Spin15s
        printf("hello, I am child (pid:%d)\n", (int)getpid());
    }
    else
    {
        // parent goes down this path (original process)
        // sleep(10);

        Spin(10);
        printf("hello, I am parent of %d (pid:%d)\n",
               rc, (int)getpid());
    }
    return 0;
}
```

运行结果如下：

```
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ gcc -o p1 p1.c
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ time ./p1
hello world (pid:15945)
hello, I am child (pid:15946)
hello, I am parent of 15946 (pid:15945)

real    0m10.001s
user    0m10.001s
sys     0m0.000s
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ gcc -o p1 p1.c
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ time ./p1
hello world (pid:16289)
hello, I am parent of 16290 (pid:16289)

real    0m10.001s
user    0m10.001s
sys     0m0.000s
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab4-process/ostep$ hello, I am child (pid:16290)
```