

系统调用-进程管理实验报告

April 4, 2023

1120213582 陈建腾

1 实验背景

1.1 实验目的

Use system calls to implement a “mytime” command to run an executable program through the command line parameter. Create a new process to run this executable program and record the running time of the new process. Implement a Linux version.

1.1.1 Hints

- Use fork()/execv() to create a new process
- Use wait() to wait for the created process to end.
- Use gettimeofday() to obtain current time.

1.2 实验环境

- VMWare Workstation 16
- Ubuntu 22.04.1 LTS
- Linux version 5.19.0-38-generic
- gcc (Ubuntu 11.3.0-1ubuntu1 22.04) 11.3.0

1.3 实验分析

1. 使用vfork()函数创建子进程，如果创建成功，子进程中返回0，父进程中返回子进程；如果创建失败，父进程中返回-1。 vfork函数保证子进程能够优先于父进程运行，在子进程exit之后父进程才会被调度运行。
2. 通过execv()函数，在子进程中调用newProcess程序。其中execv()函数中的参数需要传入二进制可执行文件，即通过gcc编译之后产生的可执行文件。execv函数传入两个参数分别为可执行文件路径和一个数组，数组的第一个参数为执行文件名字本身。
3. 使用gettimeofday()进行计时，在vfork成功进入子进程后开始计时并打印时间，在返回父进程后结束计时并打印时间，计算时间差后再次打印。

2 源代码

2.1 mytime.c

```
// mytime.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/time.h>

int main(int argc, char* argv[]){
    pid_t pid;
    struct timeval startTime, endTime;
```

```

if((pid = vfork())<0){
    printf("vfork failed!\n");
    exit(0);
}else if(pid == 0){
    gettimeofday(&startTime, NULL);
    printf("vfork succeded, now in the child process.\n");
    printf("startTime-- sec: %ld, usec: %ld\n",startTime.tv_sec, startTime.tv_usec);
    execv(argv[1], &argv[1]);
}else{
    wait(NULL);
    printf("the created process is finished, now in the parent process.\n");
    gettimeofday(&endTime, NULL);
    printf("endTime-- sec: %ld, usec: %ld\n", endTime.tv_sec, endTime.tv_usec);
    printf("the running time of the new process is:\nsec: %ld, usec: %ld\n",
endTime.tv_sec-startTime.tv_sec, endTime.tv_usec-startTime.tv_usec);
}
exit(0);
}

```

2.2 newProcess.c

```

// newProcess.c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char* argv[]){
    printf("the new process has been created.\n");
    for(int n=0; n<argc; n++){
        printf("arg[%d]:%s\n", n, argv[n]);
    }
    sleep(1);
    printf("exit the new process.\n");
}

```

3 运行与结果

3.1 运行方式

在命令行中键入

```

gcc mytime.c -o mytime
gcc newProcess.c -o newProcess
./mytime ./newProcess

```

3.2 运行示例结果

```

vfork succeded, now in the child process.
startTime-- sec: 1680581709, usec: 1680581709
the new process has been created.
arg[0]:./newProcess
exit the new process.
the created process is finished, now in the parent process.
endTime-- sec: 1680581710, usec: 180814

```

the running time of the new process is:
sec: 1, usec: 1657