

**实 验 报 告**

**（ 2019 / 2020 学年 第 1 学期）**

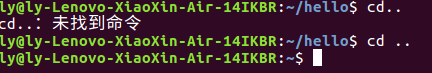
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 操作系统B | | | | | |
| 实验名称 | Linux系统及进程创建 | | | | | |
| 实验时间 | 2020 | 年 | 03 | 月 | 11 | 日 |
| 指导单位 | 计算机学院计算机科学与技术系 | | | | | |
| 指导教师 | 段卫华 | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学生姓名 | | 刘炎 | | | 班级学号 | | | B18031801 | | |
| 学院(系) | | 计软网安学院 | | | 专 业 | | | 软件工程嵌入式培养 | | |
| **实验名称** | Linux系统及进程创建 | | | | | **指导教师** | | | 段卫华 | |
| **实验类型** | 验证 | | **实验学时** | 2 | | | **实验时间** | | | 2020.3.11 |
| 1. **实验目的和要求**   1. 掌握Linux操作系统的操作和使用;  2. 掌握Linux下C语言的编辑、编译、运行的全过程;  3. 掌握进程创建系统调用的使用。 | | | | | | | | | | |
| 二、**实验环境(实验设备)**  Ubuntu系统 | | | | | | | | | | |
| **三、实验原理及内容**  **1. 学习命令并观察执行结果**  ① ctrl+L或者clear指令可以清空终端  ②ls 显示当前文件夹下的文件和目录。    ③pwd 显示当前路径的绝对路径名。 | | | | | | | | | | |

④cd改变路径

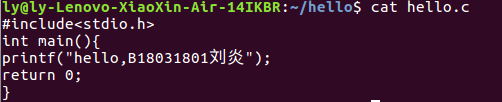
****

⑤cd..退回上一级目录

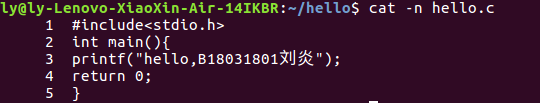


..与cd之间需要空一格

⑥cat 文件名       查看文本文件



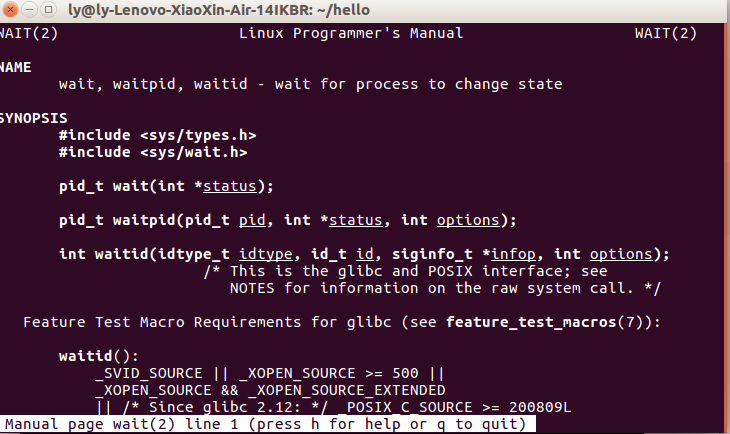
⑦cat –n 文件名    主要是-n可以显示行号

****

⑧more 文件名      类似于cat,它适合于查看大文件，进行一页一页的查看，空格键向下查看一页

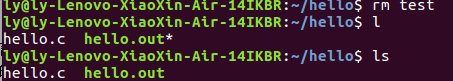
，enter键向下查看一行，按q键退出

⑨man   命令 寻求帮助命令，可以查看命令的使用手册等



⑩rm test    删除指令，删除文件test

rmdir 文件夹 删除目录

****

**2. 编辑一个C语言程序，编译运行之。**

(1) 编辑一个简单的C语言程序hello.c

#include<stdio.h>

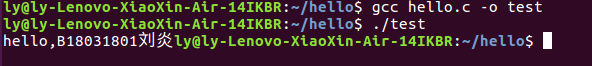
int main(){

printf("hello,B18031801刘炎");

return 0;

}

（2）编译运行该程序



gcc hello.c 是编译hello.c这个源程序，并输出可执行程序hello.out

gcc hello.c -o test是编译hello.c这个源程序，并输出名为可执行程序test

./test是运行当前路径下的可执行程序test

**3. 编写并运行创建新进程的程序，执行并观察其执行结果。**

#include <stdio.h>

int main()

{

int pid;

int i;

pid = fork();

if (pid == 0)

{

for (i = 0; i < 5; ++i)

{

printf("child %d , parentid %d\n", getpid(), getppid());

sleep(2);

}

}

else

{

for (i = 0; i < 5; ++i)

{

printf("parent %d , parentid %d\n", getpid(),getppid());

sleep(2);

}

}

wait(0);

return 0;

}

**过程：**

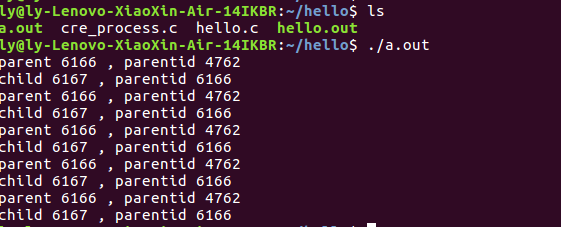
①先用mv 进行重命名。



②gcc cre\_process.c编译 ( 可以用tab进行补全)

③可以vim cre\_process.c进入进行编辑。

结果:



6166

6167

**4. 进程的同步出问题孤儿进程和僵尸进程。**

**问题：父进程比子进程先行结束，子进程成了孤儿进程。子进程先结束，但是父进程并没有释放子进程的PCB，从而导致子进程成为了僵尸进程。**

#include<stdio.h>

int main(){

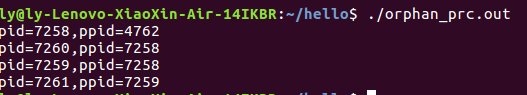
fork();

fork();

printf(“pid = %d, ppid = %d \n”, getpid(), getppid());

}

结果：



7258

7260

7259

7261

**解决：有多少个fork，就要写多少个wait与之对应。**

#include<stdio.h>

int main(){

fork();

fork();

printf(“pid = %d, ppid = %d \n”, getpid(), getppid());

wait(0);

wait(0);

return 0;

}

**僵尸进程**

#include <stdio.h>

int main()

{

int pid;

pid = fork();

if (pid > 0)

{

while(1)

{

//nothing

}

}

else

{

printf("child pid = %d, ppid = %d\n", getpid(), getppid());

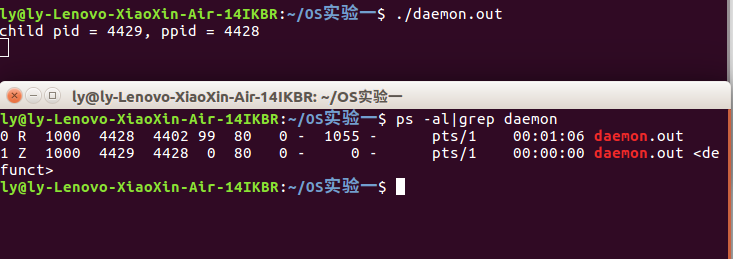
}

return 0;

}

结果：

ps -al|grep daemon 指令为



分析：

父进程死循环，子进程已经结束等待父进程进行释放PCB，成为僵尸进程。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **四、实验小结**（包括问题和解决方法、心得体会、意见与建议等）  这是操作系统B的第一个实验，通过实验熟悉了虚拟机软件VMWare的基本操作，并在虚拟机软件中熟悉了Ubuntu的基本环境。  1. 学会使用Ubuntu中的终端和文本编辑器两个软件。  2. 学会了多个Linux命令，如：pwd, ls, cd, mkdir, ps, cp, mv等等。  3. 学会了gcc编译C语言程序。  4、了解了几个个系统调用函数。  fork()  创建一个子进程。  getpid()  返回进程的进程号。  getppid()，  获得本进程的父进程号。  wait()  同步该进程的子进程，某个子进程运行结束后，该进程才会继续运行。  如果该进程没有子进程，忽略该语句。  如果该进程创建几个子进程，就应该有几个wait语句。(多点没有影响)。  sleep()  进程暂停一段时间。  **问题及其解决：**  **1、**编译报错    原因：“这个错误是由于使用了中文引号或其他全角符号，还有一种就是有中文的空格（这个不容易观察），需调到顶格处，再用tab即可。”最后发现是因为引号用的是中文符号导致。  2、对结果进行截屏时，在Ubuntu系统上不知道如何选框截屏。  解决：shift+prtsc  意见与建议： | | | | | |
| **五、指导教师评语** | | | | | |
| **成 绩** |  | **批阅人** |  | **日 期** | 2019.09.19 |