实验三 Linux环境编程及GCC、Make

**一、实验目的**

1. 掌握系统调用的实现方式并能完成基本的文件读、写，进程管理操作；
2. 掌握GCC的使用方法 ；
3. 掌握MAKEFILE的编写规则，并能使用make完成大型程序的编译、管理。

**二、实验环境**

Ubuntu

**三、实验说明**

1. 实验报告以此为模板，正文中中文字体为小四宋体，英文字体为小四Times New Roman。
2. 实验报告以“Exp2-学号-姓名”方式命名，如Exp2-201401010101-张三.doc
3. 必要时使用man等帮助命令，或网上搜索。

**四、实验内容&实验结果（包括代码和运行结果截图）**

1. 使用gcc编译fork\_CPUTime.c源文件。

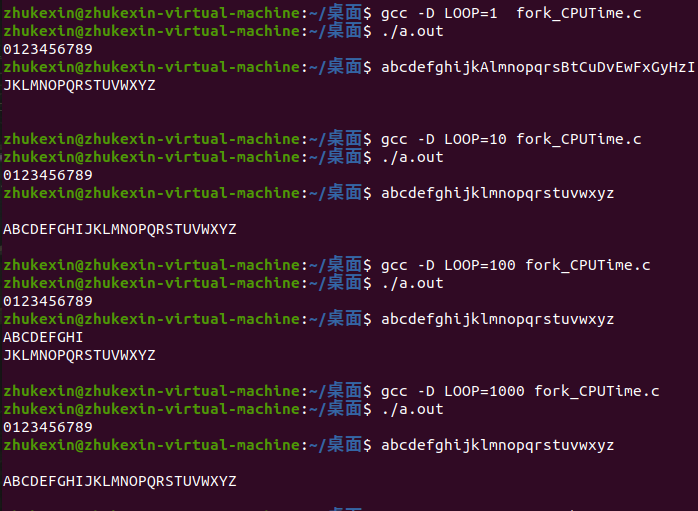
**要求：**

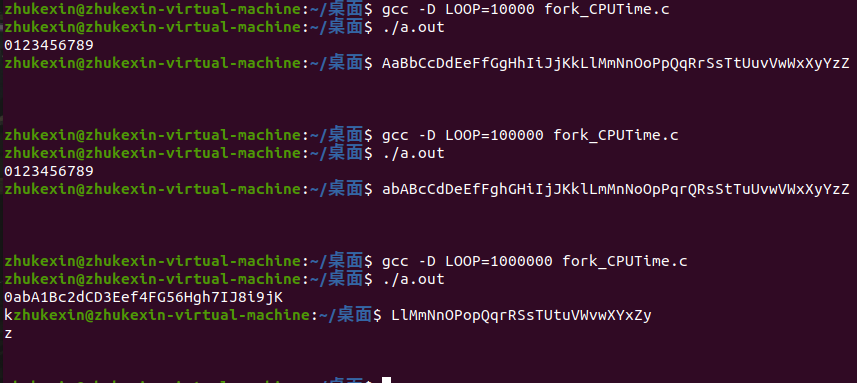
1） 通过正确的选项定义宏变量LOOP，并对其分别赋值1，10，100，1000，10000，100000，1000000；

2） 对每一次新的LOOP赋值，执行生成文件，对比结果；

3） 通过以上实例，说明Linux进程管理（CPU时间分配)的方式。

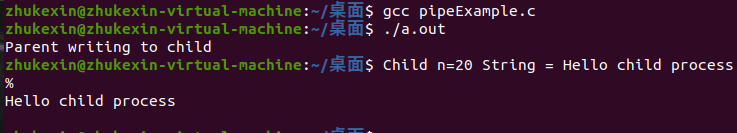
每次对新的LOOP赋值，执行生成文件：





内核在微观上，把CPU的运行时间分成许多分，然后安排给各个进程轮流运行，造成宏观上所有的进程仿佛同时在执行。双核CPU，实际上最多只能有两个进程在同时运行，我们在top、vmstat命令里看到的正在运行的进程，并不是真的在占有着CPU。

1. 使用gcc编译pipeExample.c源文件。运行生成文件，查看结果并说明文件描述符赋值规则。



Int pipe(int fd[2])代表创建管道，fd[0]代表管道读端，fd[1]代表管道写端， 若成功返回0，失败返回-1

write(fd[1],"Hello child process\n",20);//把字符串写入管道

若成功则返回写入的字节数，若出错则返回-1

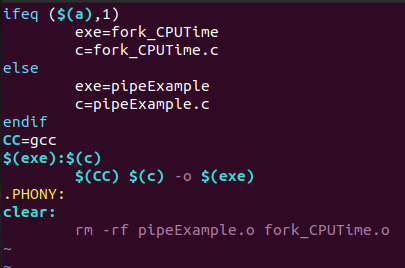
n=read(fd[0],line,21);//把管道内的内容读出来到line

若成功则返回读到的字节数，若已到文件末尾则返回0，若出错则返回-1

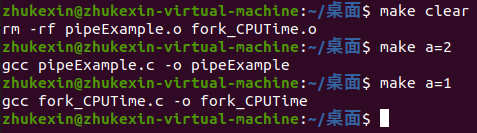
write(STDOUT\_FILENO,line,n);//向屏幕输出line

1. 编写一个makefile文件使得用户可以通过make选择编译不同的对象（如fork\_CPUTime.c或pipeExample.c）。

makefile文件内容：



通过make选择编译不同的对象：



刚开始因为格式出错，因为ifeq和左括号之间没有加空格：

IMG_256

**六、实验心得及总结**

gcc和make两种常用工具都比较实用，比较方便我们在linux上进行c++和c的编程学习，但其中很多细节需要注意，通过不断实践才能更地的学习。