实验报告

东北林业大学

信息与计算机科学技术实验中心

|  |
| --- |
| 一、实验目的  1.Vi编辑器的熟练使用  2.掌握gcc编译器的使用  3.使用fork() 函数创建子进程 |
| **二、实验环境**  实验环境：VMware, Ubuntu 22.04.5 LTS(内核版本2.6.18)  硬件环境：内存2G以上，硬盘空间20G以上 |
| **三、实验内容及结果**  （一）vi编辑器的使用  （1）使用vi命令来创建并编辑一个C语言源代码文件。  （2）使用gcc编译器来编译C语言程序  （3）运行生成的可执行文件  （二）编译并运行程序；分析程序执行一共运行几个进程，并画出进程树。  （1）使用gcc编译器来编译C语言程序sy2\_0  运行执行文件sy2\_0  得到I like linux!  进程树图为  （2）使用gcc编译器来编译C语言程序sy2\_1  运行执行文件sy2\_0  得到  pid=11227  Ppid=10956  进程树图为  （3）使用gcc编译器来编译C语言程序sy2\_2  运行执行文件sy2\_2  得到  Ppid=11419 pid=11435  Ppid=11435 pid=11436  进程树图为  （4）使用gcc编译器来编译C语言程序sy2\_3  运行执行文件sy2\_3  进程树图为  （5）使用gcc编译器来编译C语言程序sy2\_4  运行执行文件sy2\_4  进程树图为  （6）使用gcc编译器来编译C语言程序sy2\_5  运行执行文件sy2\_5  进程树图为  （7）使用gcc编译器来编译C语言程序sy2\_6  运行执行文件sy2\_6  进程树图为  （三）写出可能的实验结果及对结果进行分析说明  int main(){  int x=5;  if(fork()==0){  x=x+3;  printf("ppid=%d pid=%d x=%d\n",getppid(),getpid(),x);  }else{  x=x-3；  printf("ppid=%d pid=%d x=%d\n",getppid(),getpid(),x);  }  printf("ppid=%d pid=%d x=%d\n",getppid(),getpid(),x);  }  程序的输出取决于父进程和子进程的执行顺序。可能有两种输出结果   1. 父进程先执行   X=2 X=2 X=8 X=8   1. 子进程先执行   X=8 X=8 X=2 X=2 |
| **四、实验过程分析与讨论**  （一） vi编辑器的使用  在本实验中，Vi 编辑器用于编写 C 语言代码，在终端中输入 vi 文件名.c 打开一个新的或已有的 C 文件。按下 i 键进入插入模式，开始编写代码。使用 Esc 键退出插入模式，并输入 :wq 保存并退出编辑器。我学会了如何使用 GCC 编译器将源代码转化为可执行文件  ，当代码中出现拼写错误或语法问题时，GCC 会报错并停止编译。  （二）编译并运行程序；分析程序执行一共运行几个进程，并画出进程树。  （1） sy2\_0.c  程序只包含输出语句，没有使用 fork() 创建子进程。进程数为1 个（父进程）。程序由终端或 shell 启动后，作为一个独立的进程运行，运行后直接退出。  （2） sy2\_1.c  程序打印当前进程的 PID 和父进程的 PID，进程数为1 个（父进程）。程序未调用 fork()，只有一个进程在运行。  3. 程序 sy2\_2.c  调用 fork() 创建一个子进程。进程数为2 个（父进程和一个子进程）。fork() 调用后，子进程和父进程分别进入自己的执行路径，各自运行独立的代码。  4. 程序 sy2\_3.c  通过多次调用 fork()，共创建了 4 个进程。进程数为4 个。每次 fork() 调用会产生新的子进程，因此最终形成了树状结构的 4 个进程。  5. 程序 sy2\_4.c  两次调用 fork()，最终生成了 4 个进程。进程数为4 个。每次 fork() 会在当前进程的基础上创建一个新进程，因此形成了 4 个独立的进程。  6. 程序 sy2\_6.c  调用一次 fork()，生成一个子进程。进程数为2 个（父进程和一个子进程）。父子进程各自拥有独立的变量副本，修改 x 不会相互影响  （三）写出可能的实验结果及对结果进行分析说明  （1）父进程先执行  父进程执行时：  第一次打印：ppid=父进程 pid=子进程1 x=2，表示父进程修改了 x 的值为 2。第二次打印：重复打印当前进程中的变量 x=2。  子进程随后执行：  第一次打印：ppid=父进程 pid=子进程2 x=8，表示子进程修改了 x 的值为 8。第二次打印：重复打印当前进程中的变量 x=8。  （2）子进程先执行  子进程先执行：  第一次打印：ppid=父进程 pid=子进程2 x=8，表示子进程修改了 x 的值为 8。第二次打印：重复打印当前进程中的变量 x=8。  父进程随后执行：  第一次打印：ppid=父进程 pid=子进程1 x=2，表示父进程修改了 x 的值为 2。第二次打印：重复打印当前进程中的变量 x=2。 |
| 五、指导教师意见  ☐ 实验完成内容完整，分析及讨论充分，写作规范  ☐ 实验完成内容基本完整，分析及讨论基本充分，写作基本规范  ☐ 实验完成内容不完整，分析及讨论不充分，写作不规范  指导教师签字：侯畅  **2024年12月4日** |