|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学校** | 珠海科技学院 |  | **班级** | **22级4班** |  | **科目** | 操作系统 |
| **学院** | 计算机学院 |  | **姓名** | 陈俊伟 |  | **日期** | 2024.5.17 |
| **专业** | 计算机科学与技术 |  | **学号** | 04220411 |  | **实验名称** | 进程通信-共享内存机制 |

|  |
| --- |
| 实验内容  第1题 深度学习find和grep组合命令  命令   * 查找file.txt中含os字母的行，os无论大小写都能匹配并输出   grep -i os file.txt   * 查找file.txt中含os小写字母的行，并输出   grep os file.txt   * 查找file.txt中含OS大写字母的行，并输出   grep OS file.txt   * 查找file.txt中不含os小写字母的行，并输出   grep -v os file.txt   * 查找file.txt中不含OS大写字母的行，并输出   grep -v OS file.txt  find和grep的组合命令   * 在当前目录下所有文件中查找内容包含 string 的文件:   find ./ -name "\*" -exec grep "string" {} \;   * 在当前目录下所有文件中查找内容包含 string 的文件并列出字符所在的文件:   find ./ -name "\*" -exec grep -l "string" {} \;   * 在当前目录下 \*.c 中查找内容包含 string 的文件并列出字符所在的文件的所在行(不显示文件名):   find ./ -name "\*.c" -exec grep -n "string" {} \;   * 在当前目录下所有文件中查找内容包含 string 的文件并列出字符所在的文件,所在行及所在行的内容:   find ./ -name "\*" -exec grep -n "string" ./ {} \;  第2题 在/tmp目录下建立文件file.txt。并写入字符串“I love os”到文件中。使用find组合命令，查找/tmp目录下的所有普通文件。  命令   * 建立文件file.txt   touch /tmp/file.txt   * 通过命令行写入I love os到文件中   echo "I LOVE OS">>file.txt   * 查找/tmp目录下的所有普通文件   find -type f  运行结果截图    图1-1  第3题查找/tmp目录下文件中有“os”字符串的文件名称。  命令 find ./ -name “\*.txt” -exec grep -l “os” {} \ ;  运行结果截图    图2-1  第4题 编写一个程序，使用系统调用shmget(), shmat(), shmdt(), shmctl()实现进程通过共享内存的通信：一个子进程SERVER将共享内存首字节置为-1，当首字节值不为-1时读取这个值并再次将其置为-1；另一个子进程CLIENT在首字节为-1时将首字节置为9-0中的一个数字。(SHMKEY取值75，共享区的长度取1024)  代码  #include <stdio.h>  #include <sys/types.h>  #include <sys/msg.h>  #include <sys/ipc.h>  #include <stdlib.h>  #include <sys/wait.h>  #include <unistd.h>  #include <sys/shm.h>  #define SHMKEY 75  int shmid,i;  int \* addr;  int CLIENT()  {  int i;  shmid=shmget(SHMKEY,1024,0777);  addr=(int \*)shmat(shmid,0,0);  for(i=9;i>=0;i--)  {  while(\*addr!=-1);  printf("(client)sent\n");  \*addr=i;  }  exit(0);  }  int SERVER()  {  shmid=shmget(SHMKEY,1024,0777|IPC\_CREAT);  addr=(int \*)shmat(shmid,0,0);  do  {  \*addr=-1;  while(\*addr==-1);  printf("(server)received\n");  }while(\*addr);  shmctl(shmid,IPC\_RMID,0);  exit(0);  }  int main()  {  while((i=fork())==-1);  if(!i)  SERVER();  while((i=fork())==-1);  if(!i)  CLIENT();  wait(0);  wait(0);  return 0;  }  运行结果截图    图3-1 |
| 通过本次实验的学习，深度学习了find和gred的组合命令，以及学会编写程序，来实现共享内存机制. |