四 川 大 学 计 算 机 学 院、软 件 学 院

实 验 报 告

学号：2022141461109 姓名：\_殷浩杨\_ 专业：\_软件工程\_ 班级：\_05\_ 第 五 周

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 操作系统课程设计 | 实验课时 | 2 |
| 实验项目 | Linux进程编程 | 实验时间 | 2023/10/11 |
| 实验目的 | 1. Linux进程编程初步 | | |
| 实验环境 | ARM64，MacOS, Parallels Desktop 19，Ubuntu Linux 22.04.2 | | |
| 实验内容（算法、程序、步骤和方法） | **理解并补全下列程序：**  **fork\_execve.c**  #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  int main(int argc, char\* argv[])  {  if (/\*fork\*/) {  printf("I'm child process.\n");  /\* call 'execve' system call  \* int execve(const char \*filename, char \*const argv[], char \*const envp[]);  \* you can use 'man execve' command to get more information  \*/  char \*args[] = {"/bin/ls", NULL};  /\*execve\*/  perror("Call execve failure!!\n");  /\*exit\*/  } else {  printf("I'm parent process.\n");  /\*exit\*/;  }  return 0;  }  **fork\_execve\_wait.c**  #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <sys/types.h>  #include <sys/wait.h>  int main(int argc, char\* argv[])  {  pid\_t pid;  int status;  int options;  if (/\*fork\*/) {  printf("I'm child process.\n");  sleep(2);  /\* call 'execve' system call  \* int execve(const char \*filename, char \*const argv[], char \*const envp[]);  \* you can use 'man execve' command to get more information  \*/  char \*args[] = {"/bin/ls", NULL};  /\*execve\*/  perror("Call execve failure!!\n");  /\*exit\*/  } else {  /\*wait\*/  /\*  \* you may use wait(NULL) to get the same effect  \*/  //wait(NULL);  printf("I'm parent process.\n");  /\*exit\*/  }  return 0;  } | | |
| 实验结果的截图 | **Task1---对编写的makefile截图**  我把它分了两个文件夹放，写了两个Makefile文件    **Task2---补全源代码并截图**    **Task3---用进程命令查看并截图** | | |
| 小 结 | 通过本实验，我掌握了：   1. Linux进程编程初步   目前我存在的问题有：   1. 对进程编程理解不够深 | | |
| 指导老师评 议 | 成绩评定： 指导教师签名： | | |