**深 圳 大 学 实 验 报 告**

**课程名称： 操作系统**

**实验项目名称： 大作业Part1-1**

**学院： 计算机与软件工程学院**

**专业： 软件工程**

**指导教师： 杜智华**

**报告人：孔启楠 学号：2020155036 班级：软件工程腾班**

**实验时间： 2023.3.23**

**实验报告提交时间： 2023.3.23**

**教务部制**

|  |
| --- |
| **实验目的与要求：**  **实验目的：**  (1)、掌握计算机操作系统管理进程、处理机、存储器、文件系统的基本方法。  (2)、了解进程的创建、撤消和运行，进程并发执行；自行设计解决哲学家就餐问题的并发线程，了解线程（进程）调度方法；掌握内存空间的分配与回收的基本原理；通过模拟文件管理的工作过程，了解文件操作命令的实质。  (3)、了解现代计算机操作系统的工作原理，具有初步分析、设计操作系统的能力。  (4)、通过在计算机上编程实现操作系统中的各种管理功能，在系统程序设计能力方面得到提升。  **实验要求：**  1.阅读user/sh.c (L168-169).if(fork1() == 0) 为什么fork1()返回值为0时才进入if语句内部？  **答：应用程序sh是命令行程序，当输入一个命令时，命令行会根据命令调用fork1创建子进程（cd 命令除外），而父进程fork1返回子进程pid，子进程中fork1返回的是0，因此只有子进程才能通过条件判断，从而进行runcmd（运行命令对应的程序）**  2.阅读user/sh.c (L168-169).阅读runcmd的代码，其中：$echo README对应的cmd->type是哪个？相应的，$ls; echo “hello world“ 对应的cmd->type是哪个？而ls | wc 对应的cmd->type是哪个？给出你的答案，并从代码中给出解释。  **答：$echo README对应的cmd->type是EXEC。代表普通的可执行命令，即不包含管道符| ，后台运行符号& ，重定向符<>, 列表运行符; 的命令 。而$ls; echo “hello world“因为包含符号; 所以cmd-type=LIST 。ls | wc 对应的cmd->type是PIPE。在调用runcmd之前会先调用parsecmd来解析命令的类型和参数。解析命令的流程框架如下：**    3.子进程关闭文件描述符0后， 我们可以保证 open 会使用0作为新打开的文件 input.txt 的文件描述符（因为0是 open 执行时的最小可用文件描述符） 。 之后 cat 就会在标准输入指向 input.txt 的情况下运行。新分配的文件描述符总是当前进程中最小的未使用描述符。问题：阅读user/sh.c (L83)对应的switch分支及相关代码，请说明这个输出重定向命令$ls > test.txt如何确保test.txt接收ls命令的输出呢？  **说明：**  （1）本次实验课作业满分为100分，占总成绩的比例（待定）。  （2）本次实验课作业截至时间2022年4月3日（周日）23:59。  （3）报告正文：请在指定位置填写，本次实验不需要单独提交源程序文件。  （4）个人信息：WORD文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。  （5）提交方式：请在BLACKBOARD平台中按时提交；延迟提交不得分。  （6）发现抄袭（包括复制&粘贴整句话、整张图），该次作业记零分。  （7）期末考试阶段补交无效。 |
|  |

|  |
| --- |
| **(1).** 在xv6环境下开发应用程序  题目：仿照echo，写一个命令echo\_reversal，实现以下功能：把输入的每个参数中的字符次序颠倒输出。  例如：$ echo\_reversal Hello World!  则输出： olleH !dlroW  首先分析xv6源代码目录的结构，主要包括：kernel内核代码，user用户代码，Makefile构建脚本。本题目的echo\_reversal程序应该在是用户层面的命令，因此需要在user文件夹编写代码    模仿echo.c，编写echo\_reversal.c,其中main函数的两个参数argc和argv分布代表，命令的参数个数（以空格分割），以及每个参数具体的字符串值    代码编写完成后，需要修改Makefile,在此文件中加入本源代码文件，让其参与编译。如图    最后在命令行输入make qemu即可编译系统镜像并进入xv6系统。输入命令，即可对字符串进行反转输出。    **(2).** 回答以下问题：  Xv6中并发进程有几种状态，在源码中分别以什么常量代表，试解释每种状态的意义。  一共有6种状态，包括UNUSED, USED, SLEEPING, RUNNABLE, RUNNING, ZOMBIE这6种状态。即未使用态、初始态、等待态、就绪态、运行态、僵尸态。  在源码中的定义如下：     * UNUSED，当进程被创建但被使用时的状态，此时进程未被分配PID * USED，进程被初始化allocproc()后的状态。 * SLEEPING，进程需要等待资源时的状态 * RUNNABLE，就绪态，此时进程未被执行但是即将被执行。 * RUNNING，进程正在运行 * ZOMBIE，子进程以执行完毕，但父进程没有。此时子进程就是僵尸态   各个状态之间的转化图如下：    Xv6中PCB是以什么方式存放的，链表还是数组？系统最多允许同时运行多少个进程？  PCB的结构体的定义如下：    PCB是以数组形式存放的：    NPROC即为最大的进程数，如图，NPROC=64，即最大进程数为64：    Xv6是否支持多核cpu? 如果支持的话，是通过哪个数据结构支持的？  支持多核CPU。通过结构体”cpu”来支持，此结构体记录了该cpu运行的进程PCB，以及寄存器的值context。    另一个结构体spinlock，是同步共享资源的锁，定义如下。在xv6中，自旋锁是一种用于同步多个进程访问共享资源的锁机制。自旋锁可以用于保护对共享数据结构的访问，以确保在多个进程之间访问这些数据结构时不会出现竞争条件。自旋锁的作用是确保**同一时间只有一个进程可以获得锁并访问共享资源，其他进程需要等待当前进程释放锁后才能获得锁并访问共享资源**。当一个进程获得了自旋锁并且在访问共享资源时，其他进程**会一直自旋等待直到该进程释放了锁**。**自旋锁的好处是避免了进程进入睡眠状态，从而提高了系统的性能**。但是，如果一个进程持有自旋锁的时间过长，其他进程就会在自旋等待期间浪费大量的CPU时间。因此，自旋锁适用于短期占用共享资源的情况。    系统启动的第一个进程，其入口函数在哪个文件第几行？它主要实现什么功能？（提示：阅读《xv6中文文档》第1章“第一个进程”）  在main函数中，调用了userinit来创建第一个进程，userinit定义在(kernel/proc.c:212),  userinit调用一个initcode.S汇编程序，这个汇编程序调用exec(“/init”)，exec是通过给定参数加载并执行一个文件，这里执行的是user/init.c，在此会运行shell程序。  \*init.c中main函数，通过fork创建一个子进程，然后子进程通过exec加载shell程序文件的代码，然后子进程执行代码，父进程则等待子进程的执行。一直重复。  ++++++++++++++++++++++++++++++++++++++++++++++++++++++  **其他（例如感想、建议等等）。**  xv6 是一个精简的 Unix 操作系统，是 MIT 的操作系统课程（6.S081）的一部分。它的代码相对较小，因此非常适合用作操作系统学习的材料。学习 xv6 让我了解了操作系统的内部工作原理，以及操作系统如何管理计算机硬件和软件资源。同时，学习 xv6 还可以帮助我提高 C 编程技能和调试能力。在学习 xv6 时，我熟悉了操作系统的相关概念，例如进程、线程、调度、内存管理等等。最后，xv6 的源代码非常清晰，注释也非常详细，阅读和调试起来非常舒服 |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：谭舜泉  2022年 4 月 10 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。