# 实验一-xv6 下简单工具的编写

本文为草稿，<>中为待补充内容

## 一、实验目的

通过编写xv6系统下的简单工具，熟悉课程实验环境，复习巩固有关系统调用的知识。

## 二、预备知识

1. C 语言编程
2. Linux 系统的基本操作
3. GNU 工具链的基本使用方法
4. QEMU的基本使用方法
5. 系统 shell 的使用方法
6. 系统调用相关知识

## 三、实验环境

预置实验环境

## 四、实验内容

### 1.你好世界(简单)

#### 题目信息

难度：简单

分数：35 分

所需时间：25min

#### 题目介绍

本题目为本课程实验的第一个题目，我们希望你能通过完成本题目来熟悉课程实验环境。这个题目中附带了一份教程，此教程将帮助你迈出操作系统课程实验的第一步。

教程中将手把手教你完成如下流程

1. 打开实验目录
2. 找到需要编写代码的文件
3. 编写符合题目要求的代码
4. 保存并编译代码
5. 通过 QEMU 加载内核镜像
6. 测试代码功能
7. 提交代码至评测系统你
8. 查看成绩与测试点通过情况

在你参考教程完成上述流程后，将掌握本课程实验的实验环境与成绩系统的使用方法。

### 题目要求

本题目要求你在 xv6 系统内核中编写一段程序，使其能够输出 HelloWorld 字符串。为确保评测程序能正确评测你编写的程序，你应在`lab1/user/hw.c:5 main()`中完成本题目的代码的编写。

### 运行效果

```
(在 xv6 shell 中)
$ hw
Hello World!
```

### 教程

<此处放置前文提到的教程，最好详细些，配图要够多>

## 2.睡眠函数(简单)

### 题目信息

难度：简单

分数：40 分

所需时间：20min

### 题目介绍

在xv6 中实现类 UNIX 系统中常见的系统工具「sleep」。通过完成这个题目，你将学习到 xv6 下调用系统调用和获取命令行参数的方法。

### 题目要求

编写一个名为sleep的程序，程序运行后让当前进程休眠一定的时间。程序通过输入的命令行参数来指定休眠时间的长度，长度单位为Tick（Tick是xv6内核中定义的一个时间长度，即CPU定时器发生两个时钟中断之间的时间长短）。

为确保评测程序能正确评测你编写的程序，你应在`lab1/user/sleep.c`中完成本题目的代码的编写。

### 提示

1. 在大部分操作系统的shell中，执行可执行文件（或者说 执行命令）时，可以向可执行文件传递参数，语法为

`可执行文件名称 参数1 参数2 参数3`

比如

```
gcc a.c -o a.out
```

这条shell指令就将会执行名为gcc的可执行程序，并向其传递四个参数，参数1为 a.c，参数2为 -o，参数3为 a.out。（参数0为可执行文件自身的路径，由shell自动完成传递）
在C语言中，我们可以通过读取main函数的参数来获取当前可执行文件的参数。

```
int main(int argc, char **argv)
```

其中 main 的第一个参数 argc 为参数数量，第二个参数 argv 为指向参数字符串数组指针的指针。我们可以对其索引来获取某一个参数，例如，下列语句获取了第一个参数

```
char* arg1 = argv[1];
```

atoi函数可以被用来将一个由数字组成的字符串转换为int类型，C语言基本库中也有此函数
基本用法为

```
//输入为“2333”的C字符串
char* str = “2333”;
int num = atoi(str);
//此时num中存储着值为2333的整数
```

2. 所有操作系统都会为用户程序提供用于与操作系统交互的接口，系统调用是最常见的接口之一。用户程序可以通过系统调用来控制操作系统为用户程序做一些事情，例如创建新进程、休眠当前进程、打开新文件等等等等。Unix及其衍生的系统（比如Linux、macOS、iOS）都支持名为POSIX的系统调用集，其中包含了很多很方便的系统调用，比如open()可以用来打开文件，pipe()用来创建匿名管道，sleep()用来让当前进程休眠一段时间。
   在C语言中，我们也可以在编写xv6用户程序时调用xv6所提供的系统调用（在程序开头处需要`#include “user/user.h”`以引入系统调用的函数定义头文件）。引入后，我们就可以在程序中调用对应的函数了。
   在xv6的user/user.h中我们可以查找到这些系统调用的函数定义。其中sleep系统调用的函数定义为
   `int sleep(int);`
   其接受一个整型参数，这个参数指明了要休眠的tick数。使用方法示例：

```
sleep(5);
//当前进程休眠 5 ticks
```

3. 在你编写的程序末尾处，**必须**要通过调用exit()函数（系统调用之一）来显式地退出，否则程序会一直向下跑直到**跑飞**为止。

4. 你的程序应足够的健壮。对于不符合规范的输入，你的程序应给出适当的报错并及时地退出。
5. 如果没有思路，你可以阅读user目录下xv6内置的程序代码（例如kill.c）并从中获取灵感。

### 运行效果

```
(在 xv6 shell 中)
$ sleep 5
（一段时间内无事发生）
$
```

## 3.乒乓回响(中等)

### 题目信息

难度：中等

分数：15 分

所需时间：25 min

### 题目介绍

在xv6内核中利用匿名管道来实现两个进程之间的通信。

### 题目要求

本题目要求你在 xv6 系统内核中编写一段程序，利用系统调用创建一对管道，在两个进程之间发送并回复一个字节（类似网络编程中ping操作那样），两个进程之间互相各进行一次这样的操作。

在程序开始运行后，父进程应向子进程发送一个字节，子进程通过管道读取到这个字节后应输出"<pid>: received ping"，其中<pid>是它的进程ID，然后将管道上的字节写给父进程，随后退出；父进程应该从子进程读取字节，并输出"<pid>: received pong"，然后退出。为确保评测程序能正确评测你编写的程序，你应在 `lab1/user/user/pingpong.c` 中完成本题目的代码的编写。

### 运行效果

```
(在 xv6 shell 中)
$ pingpong
4: received ping
3: received pong
$
```

### 提示

1. 使用 pipe() 系统调用来创建匿名管道
2. 使用 fork() 系统调用来创建一个子进程。
3. 使用 read() 从管道中读取数据，使用 write() 向管道中写入数据。
4. 使用 getpid() 获取当前进程的进程ID。
5. xv6上的用户程序有一组有限的库函数（系统调用）可供你使用。你可以在user/user.h中找到这个列表。
6. 可以通过阅读 xv6 book 来学习管道的使用方法。

## 4.文件搜索(中等/困难)

### 题目信息

难度：中等/困难

分数：10 分

所需时间：35 min

### 题目介绍

在本实验中，你将实现一个简化版的 find （UNIX中查找文件的工具），通过完成这个实验，你将学会如何在UNIX系统中进行文件和目录操作。

### 题目要求

编写一个简单版本的UNIX find程序：在一个目录树中找到所有名称与指定字符串相匹配的文件。为确保评测程序能正确评测你编写的程序，你应在`lab1/user/user/pingpong.c`中完成本题目的代码的编写。

### 提示

1. 阅读 user/ls.c，学习如何读取目录项和文件。
2. 可以仿照 user/ls.c 中的文件操作来完成本实验题目代码的编写
3. 使用递归来完成对多级子目录的查找。
4. 进行递归查找时要忽略“.”（当前目录的路径）和“..”（上一目录的路径），以避免无限递归。
5. 对文件系统的改变会在qemu运行期间持续存在；如果想将文件系统初始化，则应先运行make clean，然后再运行qemu。
6. <这个实验的提示还不够，后期需要再进行补充>

### 运行效果

```
(在 xv6 shell 中)
$ mkdir a
```

```
$ echo > a/b
$ find . b
./a/b
$
```

## 关于题目难度的补充说明（实验指导书发布时可以删除本章节）

题目分为「简单」、「中等」、「困难」三个难度等级。

1. 其中「简单」难度的题目分数约占实验总分的 75%，学生在掌握题目所需的知识的前提下，仅需查阅少量的资料，即可完成「简单」难度的实验题目；
2. 其中「中等」难度的题目约占实验总分的 15%，学生在掌握题目所需的知识的前提下，需要查阅一些资料，并对代码的实现方式进行一定的思考，才能做对此难度的实验题目；
3. 其中「困难」难度的题目约占 10%，此难度的题目是为学有余力，动手能力较强，资料查找能力较强的学生准备的，学生需要完全掌握题目所需的知识，并且愿意阅读课外扩展资料，具有较强的代码设计与编写能力，才能完成困难难度的实验题目。因此困难题目所占分数权重较低。