assignment

lab

Lab Requirements

Read carefully the following project guidelines, or your project may be rejected and you receive a grade of 0

Use the template `os_project_X_Y_Z_2021Fall`, which is the **project name**. Replace the "X", "Y", and "Z" with the project number (start from 1 and increase 1 per project), your cqut id, and you name in `CamelCase` in `English` (ChengLin for example, family name first), repectively. I ask you to write your name in English because Chinese characters become wrong-encoded on my computer sometimes.

Put all program-related files in the `code` folder. If you have multiple files, put them side by side. Do not make sub-folders under the `code` folder.

Write a report and put it in the `report` folder. The report must be in `PDF` format. Your report must be well-structured and thus easy to read. Your report must indicate:

1. all scripts or commands to compile, run your code. You should also write this part in the **readme.md** file
2. your project preparation, design, and result
3. necessary tables, charts
4. screenshots of output
5. a summary of what you have learned from the project
6. citations if you use materials from other source
7. if you work in groups, indicate each one's contribution
8. other necessary infomation

Make a `.zip` file of your folder. Do not use other zip format such as .rar. The name of the zip file must be `project name.zip`. Send you project to `chenglin@cqut.edu.cn`. The email subject must be the `project name`.

Lab 1: Linux Utilities

Finish the **wcat** and **wzip** project.

Lab 2: Process API

Finish `Shell` project.

Lab 3: Thread API

Problem Statement

Five silent philosophers sit at a round table with bowls of noodls. Chopsticks are placed between each pair of adjacent philosophers.

Each philosopher must alternately think and eat. However, a philosopher can only eat noodle when they have both left and right chopsticks. Each can be held by only one philosopher at a time and so a philosopher can use the chopstick only if it is not being used by another philosopher. After an individual philosopher finishes eating, they need to put down both chopsticks so that the chopsticks become available to others. A philosopher can only take the chopstick on their right or the one on their left as they become available and they cannot start eating before getting both chopsticks.

Eating is not limited by the remaining amounts of noodle or stomach space; an infinite supply and an infinite demand are assumed.

Project

In this project, you will write concurrent programs to observe and address deadlocks.

Write three programs: `dp1.c`, `dp2.c`, and `dp3.c`. In `dp1.c`, implement a naive algorithm to observe deadlock happen. In `dp2.c` and `dp3.c`, improve `dp1.c` in two different ways to address the deadlock. That is, the philosophers can keep eating. Choose any two algorithms that work, and remember to explain them in your report. All three programs take no input, and print to `STDOUT` the following information: process id, thread id, timestamp, eating count for each philosopher. Format the output properly such that it is easy to read. Below is a possible format, but you may create yours. Consider use `printf` with the `\t` format control. You may include other files such as C header files.

Write a `makefile` to build your project. The makefile shall have at least two rules. One is the default rule `make`, which produces three executables `df1`, `df2`, and `df3`; the other is a cleanup rule `make clean`, which deletes everything generated by the default rule.

hline pid	tid	time	p1	p2	p3	p4	p5
1	1	0	12				
1	1	0	112				
1	1	0		1000			

Lab 4: XV6 Utilities

这次实验，同学们学习一个新的操作系统：XV6. 它的主页在这里.
按如下操作。

- 克隆 XV6 代码仓库。

```
git clone git://g.csail.mit.edu/xv6-labs-2021
cd xv6-labs-2021
git checkout util
```

- 按照 Labs:tools 安装相关工具
- 尝试阅读 *XV6 book*。
- 尝试阅读 XV6 源码。找出 XV6 提供的系统调用函数。
- 在用户空间，`user` 文件夹下，添加一个函数。该函数至少使用一个系统调用。测试你的函数。
- 为 XV6 添加一个系统调用。你可以在 XV6 代码库根目录下运行下面的命令，查找一个 optime 系统调用的出现情况。

  ```
  find . -name "*" -type f | xargs grep uptime
  ```

  在用户空间添加一个函数测试你的系统调用。

Cousre Design

课程设计时间：2:00 p.m. - 5:40 p.m.
Week 17, 六; Week 18, 五; Week 19, 五 六;

这次课程设计的目的：同学们将设计和实现综合型操作系统项目，进一步提高操作系统的理解和掌握，同时提升学习能力和解决问题的能力。

课程设计的题目，同学们可以自行选择，但必须与操作系统相关，并且难度与下面的参考项目相当。如果你自己选择课程设计题目，必须在课程设计开始前把题目、实现的功能、实现方法发给我。下文有一些参考题目，你也可以直接使用。

课程设计按照分组完成。每个小组必须在课程设计开始前选择好题目，也就是你的课程设计做什么。

在课程设计结束的时候，我们会举行课程设计答辩。每个小组要展示自己的项目，并回答老师和其它同学的提问。最后，各小组需要提交课程设计作业相关文件。

根据我们课程设计的目的，希望同学们知识的进步和能力的提升。因此，在答辩和报告中，应该突出你在课程设计中学到了什么，进行了什么努力，克服了什么困难。完全实现项目最初的所有的要求并不是必要的。

建议大家早动手，因为在多个小时间段完成项目，比一次完成更容易。

- `concurrency-mapreduce` in ostep-projects
- `concurrency-pzip` in ostep-projects
- `concurrency-webserver` in ostep-projects
- Lab `system calls` 及以下的项目 XV6 Labs