IPADS INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEMS

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

课程实验4：多核

陈海波 / 夏虞斌

负责助教：胡雨奇 (yuki.h@sjtu.edu.cn)

上海交通大学并行与分布式系统研究所

https://ipads.se.sjtu.edu.cn

版权声明

- 本内容版权归**上海交通大学并行与分布式系统研究所**所有
- 使用者可以将全部或部分本内容免费用于非商业用途
- 使用者在使用全部或部分本内容时请注明来源：
  - 内容来自：上海交通大学并行与分布式系统研究所+材料名字
- 对于不遵守此声明或者其他违法使用本内容者，将依法保留追究权
- 本内容的发布采用 Creative Commons Attribution 4.0 License
  - 完整文本：https://creativecommons.org/licenses/by/4.0/legalcode

实验准备

实验获取

- 实验代码发布在公共远端仓库（upstream）下的 lab4 分支
- 使用Git操作与本地修改合并，并与自己的远端仓库（origin）同步

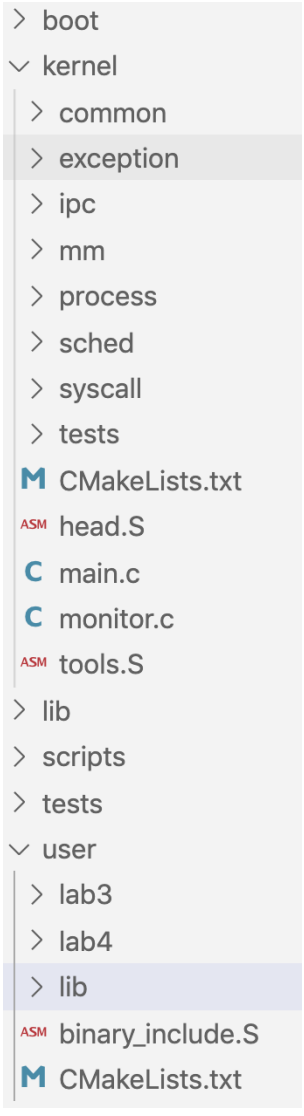
```
# commit all your previous solution of lab3
$ git commit -am 'my solution to lab3'

# fetch the remote updates, you are in branch lab3
$ git fetch upstream

# you switch to the branch lab4, whose code is based on the
# empty code provided by the TAs, you are in branch 4
$ git checkout -b lab4 upstream/lab4
Branch 'lab4' set up to track remote branch 'lab4' from 'upstream'.
Switched to a new branch 'lab4'

# you need to merge your lab1~3's solutions to lab4
$ git merge lab3
Merge made by recursive.

# commit the megre to branch lab4
$ git commit -am "merged previous lab solutions"

# update the remote tracking branch to your origin repo
# instead of the upstream repo
$ git push -u origin
To https://ipads.se.sjtu.edu.cn:2020/[username]/chcore.git
 * [new branch]      lab4 -> lab4
Branch 'lab4' set up to track remote branch 'lab4' from 'origin'.
```

注意

- **按照要求修改指定文件或函数**
- **独立完成，切勿抄袭！**
  - 账号和**个人项目**请勿泄露
- **请按时提交**
  - 鼓励多次git commit & git push

文件结构与Lab3类似

- **boot：boot代码**
- **kernel：操作系统内核代码**
  - common：kernel内部库
- **user：用户程序代码**
  - lab4：lab4的测试程序
  - lib：用户库
- **lib：boot、kernel、user共用库**



命令行指令

- **编译用户程序**
  - make user
- **编译内核**
  - make build
    - 内核测试
  - Make build bin=xxx
    - “xxx”为用户程序名
- **运行**
  - make qemu
- **调试**
  - make qemu-nox-gdb
  - make gdb

- **Lab3中的指令同样有效**
  - make run-x
  - make run-x-gdb

编译问题 – Werror

- **ChCore默认编译时开启了Werror**
  - 禁止存在任何Warning
  - 减少出现bug的概率
- **然而，实现过程中**
  - Merge后的代码框架可能会产生Warning
  - 可暂时关闭Werror（CMakeLists.txt中）
- **最终评分前请恢复Werror标签！**

注意

- **在运行内核测试时，不会指定用户程序**
  - 请忽略内核测试的“No Given Test” BUG报错
- **请注释Lab 3中 sys_exit()函数中的任何命令行输出**
  - 否则可能会导致部分Lab 4的评分出现错误
- **Lab 4共有 3questions+14exercises+1bonus**
  - 每个exercise切分很细，代码量不大

实验四简介

实验四

- **发布时间: 2021-04-14**
- **截止时间: 2020-05-21 23:59 (GMT+8)**
- **负责助教: 胡雨奇(yuki.h@sjtu.edu.cn)**
- **实验目的**
  - 将ChCore扩展为多核系统
  - 实现一个调度器
  - 从用户态运行一个新的程序
  - 实现进程间通信

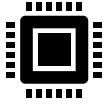
四个部分

- **Part A: Multiprocessor Support**
- **Part B: Scheduling**
- **Part C: Spawn**
- **Part D: Inter-Process Communication (IPC)**

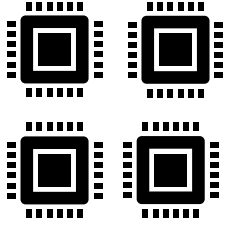
Part A – 多核

- 将ChCore从单核扩展成为多核系统

Lab1~3



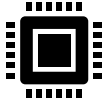
Lab4



Part A – 多核初始化

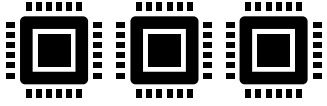
系统启动时区分：

Bootstrap Processor



- **系统初始化**
  - **uart_init()**
  - **mm_init()**
  - **启动Application Processor**
- **CPU本地初始化**

Application Processor



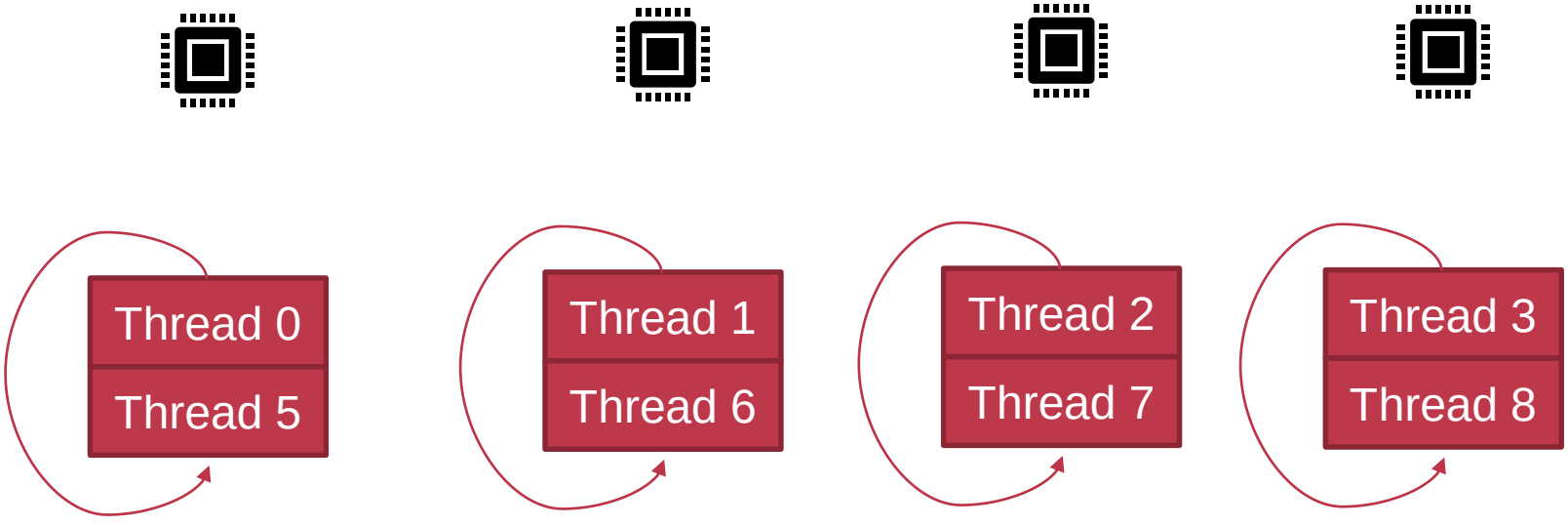
CPU本地初始化

Part A – 内核全局锁

- 使用一把全局排号锁避免多核的并发控制问题
- 理解ticket lock代码，填空
- 在用户态进入内核态的入口处**加锁**
- 在内核态返回用户态的出口处**放锁**

Part B – 调度

- 实现一个Per-CPU Round Robin 调度器



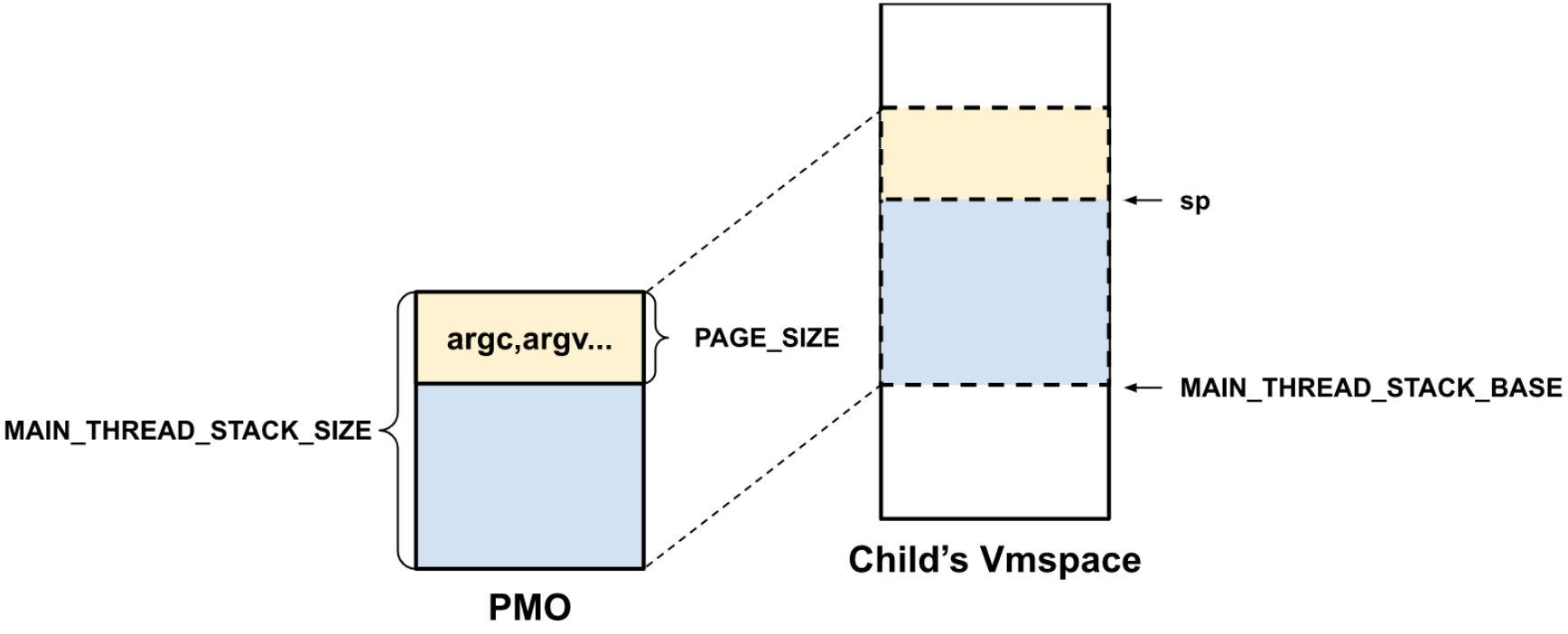
Part B – 扩展

- **抢占式调度**
  - 处理硬件时钟中断
  - 每个线程只能够执行固定时间片
- **CPU Affinity**
  - 指定调度、执行该线程的CPU

Part C – Spawn

- **创建新的进程并运行指定程序**
- **大家熟悉的（ICS）：**
  - fork() + exec(./binary)
- **我们要实现的：**
  - spawn(./binary)

Part C – Spawn

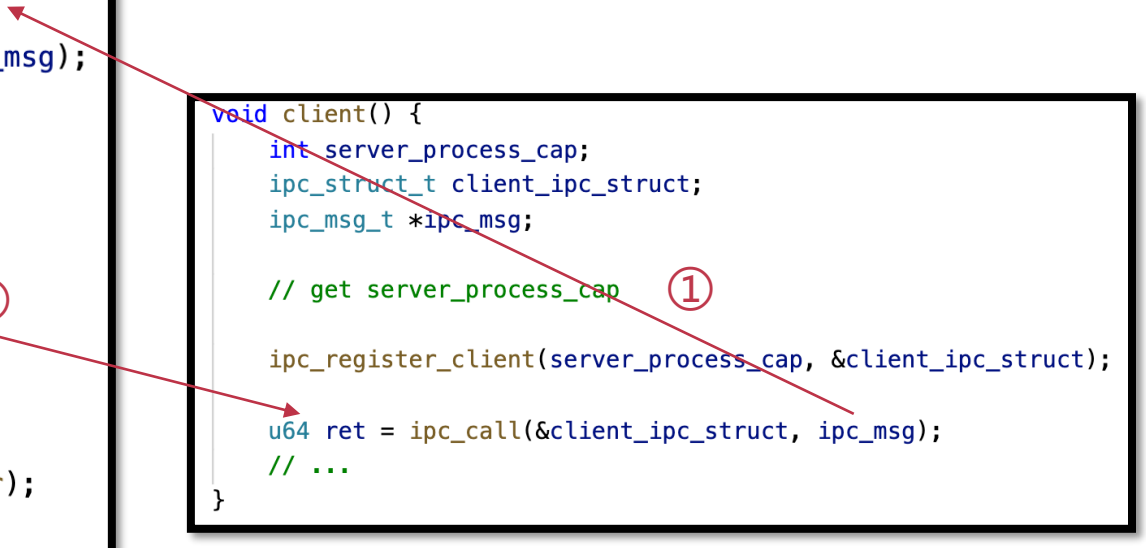


- 由于涉及Capability概念，代码填空为主

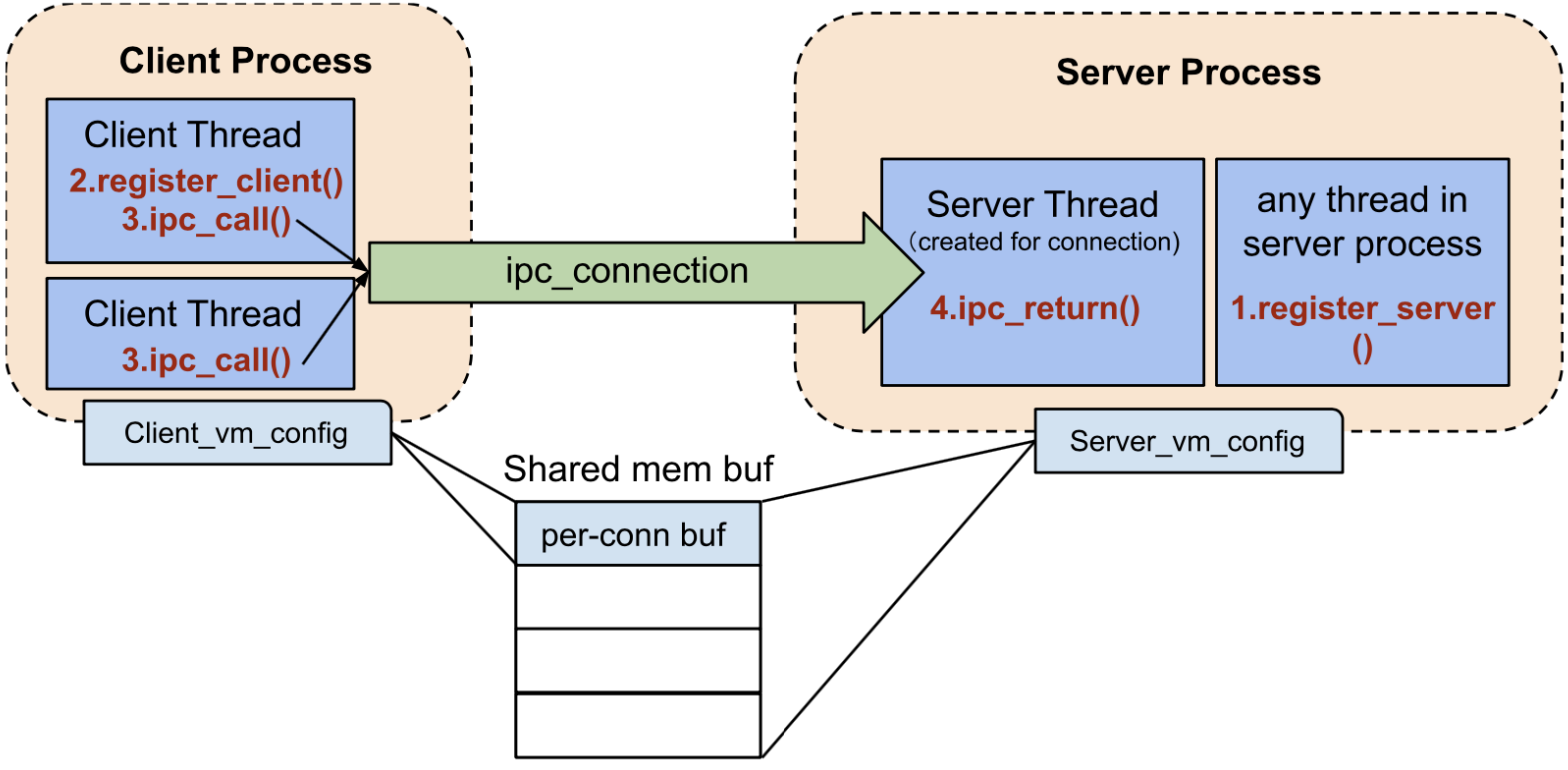
Part D – 进程间通信

```c
void ipc_dispatcher(ipc_msg_t *ipc_msg) {
    u64 ret;
    char* data = ipc_get_msg_data(ipc_msg);

    // handling ipc accordingly...

    ipc_return(ret);
}

void server() {
    // ...

    ipc_register_server(ipc_dispatcher);

    // ...
}
```

```c
void client() {
    int server_process_cap;
    ipc_struct_t client_ipc_struct;
    ipc_msg_t *ipc_msg;

    // get server_process_cap

    ipc_register_client(server_process_cap, &client_ipc_struct);

    u64 ret = ipc_call(&client_ipc_struct, ipc_msg);
    // ...
}
```



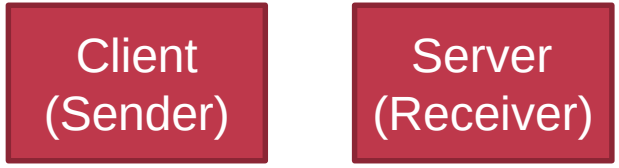
Part D – 进程间通信



- 由于涉及Capability概念，代码填空为主

Part D – 奖励部分

- 实现一个基本的、可运行的Send/Recv IPC
  - 没有接口限制、无需实现共享内存传递
  - 只需要传输64-bit值即可
- 创建一个新的Lab分支并在其上实现
  - 可以复用部分现有IPC代码
  - 自己给出一个测试用例
  - 在文档中描述关键设计



Enjoy Your Lab

- **Q&A**