Lab4 实验报告

Author: 刘佳隆

Student ID: 518010910009

Email: liujl01@sjtu.edu.cn

思考题 1

阅读汇编代码kernel/arch/aarch64/boot/raspi3/init/start.S。说明ChCore是如何选定主CPU，并阻塞其他其他CPU的执行的。

主CPU先执行`_start`函数：

首先获取CPU标识（MPIDR）并判断是否为主CPU:

- `mrs x8, mpidr_el1` 指令读取了多处理器ID寄存器（MPIDR_EL1），该寄存器包含了CPU的标识信息。
- `and x8, x8, #0xFF` 将MPIDR的低8位与0xFF进行按位与操作，通常情况下，低8位可以用来识别CPU核心编号。
- `cbz x8, primary` 如果结果为0（即CPU核心编号为0），跳转到primary标签处执行，这意味着这将是主CPU的执行路径。

在判断为主CPU之后，会跳转到`init_c`函数执行，而后通过`wakeup_other_cores`将其他核心从`wait_for_event`指令中唤醒，也开始执行`_start`函数。

非主CPU在执行`_start`函数中判断是否为主CPU之后，会进入`wait_for_bss_clear`等待`clear_bss_flag`内存位置的值变为0。而`clear_bbs()`在主CPU进入init初始化之后就会被执行，因此此处并会不阻塞，而是切换到el1并准备栈指针。当非主CPU执行到`wait_until_smp_enabled`时，会等待`secondary_boot_flag`被置为1之后才能跳转到`secondary_init_c`继续执行第二阶段的初始化。

当主CPU执行完`wakeup_other_cores`之后会继续执行`start_kernel(secondary_boot_flag)`，并在这个过程中将所有非主CPU的`secondary_boot_flag`置为1，进而使非主CPU可以继续初始化。

因此，代码首先通过CPU标识判断哪个是主CPU，主CPU直接进入初始化操作系统的流程，而其他CPU则会等待主CPU准备好SMP之后才会继续执行其特定的初始化代码。这样就实现了在系统启动时区分和控制不同CPU核心的行为。

思考题 2

阅读汇编代码kernel/arch/aarch64/boot/raspi3/init/start.S, init_c.c以及kernel/arch/aarch64/main.c，解释用于阻塞其他CPU核心的secondary_boot_flag是物理地

址还是虚拟地址？是如何传入函数enable_smp_cores中，又是如何赋值的（考虑虚拟地址/物理地址）？

`secondary_boot_flag` 物理地址

1. 在 `init_c` 函数中调用 `start_kernel(secondary_boot_flag)` 函数，传入 `secondary_boot_flag`， `start_kernel` 函数会调用 `main` 函数
2. 在 `main` 函数中调用 `enable_smp_cores(boot_flag)`，此处的 `boot_flag` 即为 `start_kernel` 传入的 `secondary_boot_flag`
3. 在 `enable_smp_cores` 函数中，由于主CPU开启了MMU，因此需要先通过 `phys_to_virt` 将 `boot_flag` 从物理地址转化为虚拟地址 `secondary_boot_flag`， `secondary_boot_flag` 指向一个数组，数组下标对应于cpu_id，再根据CPU的个数 `PLAT_CPU_NUM` 依次将从CPU的 `secondary_boot_flag` 置为1

练习 9

尝试优化在第三部分实现的IPC的性能，降低test_ipc_perf.bin的三个测试所消耗的cycle数

测试是对创建线程到所有线程运行结束的时间，通信过程可分为如下四个阶段：

1. 客户端进程通过连接的capability发起进程间通信请求
2. 内核检查权限，若通过则继续步骤3,否则返回错误
3. 内核直接切换到服务端进程执行（不经过调度器），将通信请求的参数设置给服务端进程的寄存器中
4. 服务端处理完毕后，通过与步骤3相反的过程将返回值传回客户端

对应在代码中，优化的核心部分是 `sys_ipc_call` 和 `sys_ipc_return`，优化的可能的方向有：

1. 减少进程切换开销：在 ChCore 中的 IPC 使用直接切换，对该部分的优化较困难
2. 减少不必要的检查：我们的测试整体上是较为安全的，为了追求更好的性能，一个直观的方法就是减少对特殊情况的检查
3. 减少函数调用开销：一个简单的方法就是将部分函数改为 inline，主要为 `cap_free`、`eret_to_thread`，后来发现改为静态函数需要更改地方较多且许多函数没有源码，因此直接在调用处进行修改

修改代码位置及对应上述方法为 `connection.c`：

- Line 590-591 - 方法3
- Line 607-609 - 方法2
- Line 616-620 - 方法2
- Line 648-649 - 方法3
- Line 740-778 - 方法2

由于随着树莓派的运行，`./test_ipc_perf.bin`的运行结果会越来越慢，因此使用优化前后各四次运行结果进行对比：

优化前：

```
$ ./test_ipc_perf.bin
[procmgr] Launching /./test_ipc_perf.bin...
load library name:/./test_ipc_perf.bin
map library base:0x7b92a4242000
load library complete
[procmgr] Launching test_ipc_server.bin TOKEN...
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
load library name:test_ipc_server.bin
map library base:0x77d7fec72000
load library complete
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
[TEST] test ipc with 32 threads, time: 44221982 cycles
[TEST] test ipc with send cap, loop: 100, time: 11264335 cycles
[TEST] test ipc with send cap and return cap, loop: 100, time: 22197331 cycles
[TEST] Test IPC Perf finished!
```

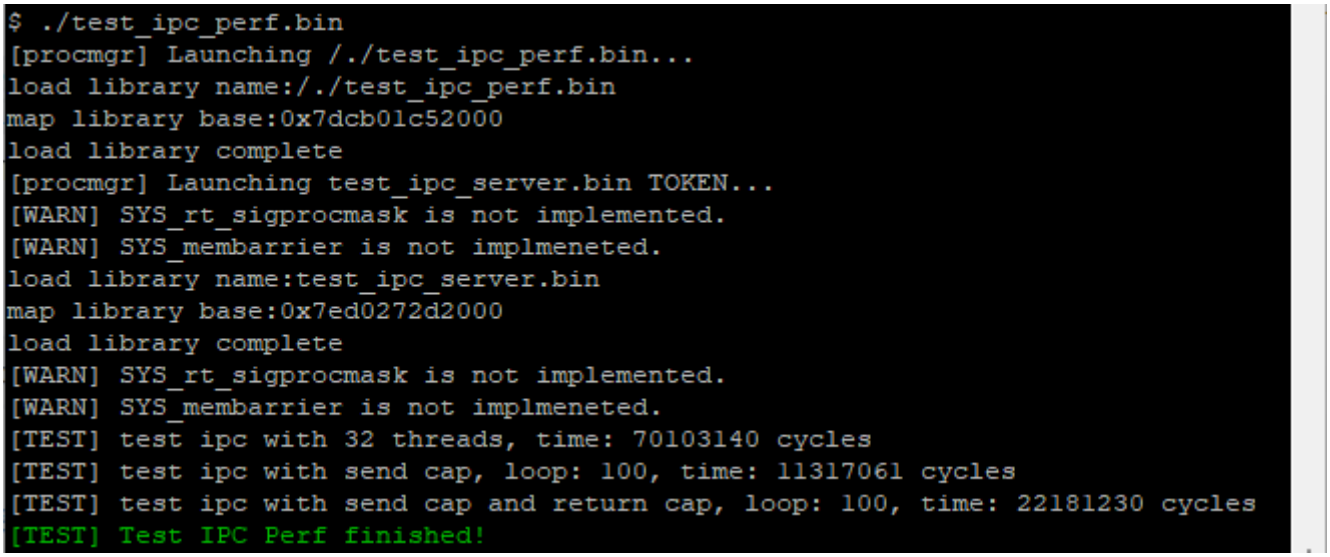
```
$ /test_ipc_perf.bin
[procmgr] Launching /test_ipc_perf.bin...
load library name:/test_ipc_perf.bin
map library base:0x7eda7ce22000
load library complete
[procmgr] Launching test_ipc_server.bin TOKEN...
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
load library name:test_ipc_server.bin
map library base:0x70eeb89e2000
load library complete
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
[TEST] test ipc with 32 threads, time: 61191992 cycles
[TEST] test ipc with send cap, loop: 100, time: 11300162 cycles
[TEST] test ipc with send cap and return cap, loop: 100, time: 22147899 cycles
[TEST] Test IPC Perf finished!

$ ./test_ipc_perf.bin
[procmgr] Launching /./test_ipc_perf.bin...
load library name:/./test_ipc_perf.bin
map library base:0x70la5c422000
load library complete
[procmgr] Launching test_ipc_server.bin TOKEN...
[WARN] SYS_rt_sigprocmask is not implemented.
load library name:test_ipc_server.bin
[WARN] SYS_membarrier is not implmeneted.
map library base:0x7de225242000
load library complete
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
[TEST] test ipc with 32 threads, time: 67915590 cycles
[TEST] test ipc with send cap, loop: 100, time: 11385760 cycles
[TEST] test ipc with send cap and return cap, loop: 100, time: 22211037 cycles
[TEST] Test IPC Perf finished!

$ ./test_ipc_perf.bin
[procmgr] Launching /./test_ipc_perf.bin...
load library name:/./test_ipc_perf.bin
map library base:0x72al8a932000
load library complete
[procmgr] Launching test_ipc_server.bin TOKEN...
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
load library name:test_ipc_server.bin
map library base:0x798844fd2000
load library complete
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
[TEST] test ipc with 32 threads, time: 88645686 cycles
[TEST] test ipc with send cap, loop: 100, time: 11387083 cycles
[TEST] test ipc with send cap and return cap, loop: 100, time: 22448131 cycles
[TEST] Test IPC Perf finished!
```

优化后：

```
$ ./test_ipc_perf.bin
[procmgr] Launching /./test_ipc_perf.bin...
load library name:/./test_ipc_perf.bin
map library base:0x76bfe08d2000
load library complete
[procmgr] Launching test_ipc_server.bin TOKEN...
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
load library name:test_ipc_server.bin
map library base:0x75584dff2000
load library complete
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
[TEST] test ipc with 32 threads, time: 52821779 cycles
[TEST] test ipc with send cap, loop: 100, time: 10945054 cycles
[TEST] test ipc with send cap and return cap, loop: 100, time: 21792860 cycles
[TEST] Test IPC Perf finished!
```

```
$ ./test_ipc_perf.bin
[procmgr] Launching /./test_ipc_perf.bin...
load library name:/./test_ipc_perf.bin
map library base:0x721b7a872000
load library complete
[procmgr] Launching test_ipc_server.bin TOKEN...
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
load library name:test_ipc_server.bin
map library base:0x7a0d8fbf2000
load library complete
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
[TEST] test ipc with 32 threads, time: 65780602 cycles
[TEST] test ipc with send cap, loop: 100, time: 11141516 cycles
[TEST] test ipc with send cap and return cap, loop: 100, time: 22065427 cycles
[TEST] Test IPC Perf finished!
```

```
$ ./test_ipc_perf.bin
[procmgr] Launching /./test_ipc_perf.bin...
load library name:/./test_ipc_perf.bin
map library base:0x704502972000
load library complete
[procmgr] Launching test_ipc_server.bin TOKEN...
[WARN] SYS_rt_sigprocmask is not implemented.
load library name:test_ipc_server.bin
[WARN] SYS_membarrier is not implmeneted.
map library base:0x72e6a83f2000
load library complete
[WARN] SYS_rt_sigprocmask is not implemented.
[WARN] SYS_membarrier is not implmeneted.
[TEST] test ipc with 32 threads, time: 69093201 cycles
[TEST] test ipc with send cap, loop: 100, time: 23107833 cycles
[TEST] test ipc with send cap and return cap, loop: 100, time: 39946638 cycles
[TEST] Test IPC Perf finished!
```



测试结果仍然很随机，似乎优化并没有啥用。