任务：TLB 例外处理和页替换

中国科学院大学 操作系统研讨课

2017.11.29

1. 介绍

本任务的主要目的是实现 TLB 例外处理。

1.1. 需要了解的部分

- MIPS虚拟内存访问流程
- TLB例外处理
- TLB操作指令

2. 初始代码

2.1. 文件介绍

- Makefile: 编译文件。
- bootblock.s: 内核启动程序，**请使用作业一中自己写的代码**。
- Createimage.c: 生成内核镜像的Linux工具，**请使用作业一中自己写的代码**。
- Entry.S: 中断处理函数。
- kernel.c: 内核最先执行的文件，放在内核的起始处。
- scheduler.c: 调度器，实现*task*的调度。
- syslib.S: 系统调用函数
- syslib.c: 系统调用接口
- interrupt.c: 系统调用和中断处理相关的函数。
- queue.c: 队列处理函数，提供了队列操作的一些接口。
- print*.c：提供一些输出函数，可以用于调试以及显示信息。
- sync.c: 一些同步操作。
- mbox.c: 邮箱的操作。
- util.c: 提供了一些输出函数，可以用于调试以及显示信息。
- file.c: 将所有任务放在File的机构体中，供do_spawn查找使用
- ramdisk.c：提供操作File的一些接口
- *.h: 相应.c/.S文件的头文件。
- Memory.c: 需要实现的关于虚拟内存的函数。
- Memory.h: 需要实现的关于虚拟内存的一些定义。

2.2. 获取

课程网站。

2.3. 运行

Makefile 文件提供编译功能。
make 编译命令
make clean 对编译产生的文件进行清除
sudo dd if=image of=/dev/sdb 将产生的 image 写进 SD 卡中
在 minicom 中执行 *loadboot* 运行程序

3. 任务

3.1. 设计和评审

帮助学生发现设计的错误，及时完成任务。学生需要对这次的作业进行全面考虑，在实现代码之前有清晰的思路。学生讲解设计思路时可以用不同的形式，如伪代码、流程图等，建议使用 PPT。

3.1.1.设计介绍

对于需要 TLB 进行虚实转换的内存区域访问，处理器会首先查询 TLB，转换成物理地址后进行实际访问。如果转换失败，会发生 TLB 例外。本次任务要求在初始化页表的时候只是分配页表项，并不分配实际的页面（即 on-demand paging 方式），只有在实际访问到该页面时才会真正分配物理页框。

当物理内存不够用时，会发生页替换，本次任务需要设计 FIFO 的页替换，同时实现 pin 页。

3.2. 开发

3.2.1.要求

本次任务主要是实现 TLB 缺失中断处理函数。
entry.S:
handle_tlb：处理 TLB 例外
memory.c:
setup_page_table：实现 on-demand paging

其他 handle_tlb 中需要的 c 函数。

本任务中需要考虑物理页面不够用时实现页替换。可以通过修改可以用的物理内存大小（物理页面个数）触发页替换。本任务中页替换算法使用 FIFO 即可。为了实现页替换，需要考虑在 page 结构中增加哪些域来维护替换信息。另外，页替换时需要实现 pin，即考虑哪些页不能替换，以及如何实现 pin。

3.2.2.注意事项

请认真阅读预备知识中的关于 TLB 的结构和操作的相关文档。

- hanle tlb

本任务中 TLB 相关的例外由 handle_tlb 处理。TLB 例外考虑两种情况：一种是 TLB 中找不到匹配项，进而查找页表(TLB miss)，页表中已有映射关系，填充 TLB 即可(任务一中实现)；另一种是 TLB 找不到匹配项，并且页表中的页表项是无效的（page fault），或者虽然 TLB 找到匹配项，但页表项是无效的（page fault）；此时需要分配新的物理页面，并更新页表项。

- tlb操作指令

本任务用到以下 TLB 操作指令，我们提到的寄存器的名字均已定义好，可以在 cp0regdefs.h 中查看，格式为：CP0_xxxxxx，如 CP0_INDEX 表示 index 寄存器。

tlbp: tlp lookup

搜索虚拟页号及 ASID 跟当前 EntryHi 中的值相匹配的 TLB 项，并把该项的索引保存到 Index 寄存器。若没有找到匹配则 Index 会被设置成负值。

示例代码：

```
li a0, 1
li a1, 0x1000000
add a1, a1, a0
mtc0 a1, CP0_EntryHi
tlbp
```

这段代码执行后，会在 TLB 中寻找 ASID 是 1，虚拟地址是 0x1000000 >> 6 的 TLB 表项，如果命中，假设在 TLB 的第 2 项命中，则 CP0_INDEX 寄存器中的值为 2，否则为负数。总之，tlbxxx 这样的指令有点类似系统调用，我们需要设置好相应的寄存器，调用命令后，结果会保存到相应的寄存器中。

tlbr: read TLB entry at index

将 index 选中的 TLB 表项的内容传送到 EntryLo0 和 EntryLo1 中。

tlbwi: write tlb entry at index

将 EntryLo0 和 EntryLo1 中的内容写到 index 选中的 TLB 表项中。

tlbwr: write tlb entry selected by random

随机在 TLB 中选择一个位置，将 EntryLo0 和 EntryLo1 中的内容写入。

- 进程的用户态栈设置

本任务中需要将进程的栈设置在用户空间（0x00000000 -- 0x7FFFFFFF），请修改相应的建栈函数，同时需要为进程栈建立页表。注意：栈页表也需要采用 on-demand paging 方式，不要像任务一那样预先分配。

- pin/unpin页

本任务中需要考虑哪些页是需要 pin 住的，即不能换出的。同时，需要考虑

如何实现 pin？

- 代码段加载

本任务中需要实现 on-demand paging，因此不需要在 setup_page_table 中预先加载代码段。你需要实现在处理 page fault 分配页面后，再拷贝代码。

- 页替换的交换区

针对页替换，可以划分出一块物理内存模拟磁盘交换区，并使用 bcopy 执行交换页在内存和模拟交换区之间的交换。

4. 测试

本任务测试时，需要确保初始化页表时页面设置为无效，其他测试过程和任务一相同。

另外，请增加显示 tlb_refill 和 page_fault 的次数。