

# ucore 在 v9-CPU 上的移植

---

韩旭 徐磊

# 目标描述

---

- 移植ucore的代码到v9-CPU上
- 实现更多的内存分配算法
- 实现更多的页面替换算法

# 实现进度

---

- 已完成：物理内存管理、虚拟内存管理、进程控制
- 即将完成：同步互斥、文件系统（debug）
- 实际完成步骤：
  - lab1-lab2 （中断，物理）
  - lab3 （虚拟）
  - lab4-lab6 （内核，用户，调度）
  - lab7-lab8

```
10 warnings generated.
./xc : compiling ucore.c
./xc : ucore.c compiled with 0 errors
entry = 12940 text = 12960 data = 1328 bss = 1096
./xc : exiting
(THU.CST) os is loading ...
memory management: default_pmm_manager
check_alloc_page() succeeded!
check_pgdir() succeeded!
check_boot_pgdir() succeeded!
----- BEGIN -----
PDE(e0) c0000000-f8000000 38000000 urw
  |-- PTE(38000) c0000000-f8000000 38000000 -rw
PDE(1) fac00000-fb000000 400000 -rw
  |-- PTE(e0) faf00000-fafe0000 e0000 urw
  |-- PTE(1) fafeb000-fafec000 1000 -rw
----- END -----
trapframe at c0003bd0
  SP : 0x36154
  REG_G : %f
  REG_F : %f
  REG_A : 0x7d00000
  REG_B : 0xc0003c08
  REG_C : 0x0
Trap in kernel!
Error Code: 1
PC : 0xc0003284
```

```
10 warnings generated.
./xc : compiling ucore.c
./xc : ucore.c compiled with 0 errors
entry = 36636 text = 36680 data = 3840 bss = 4219296
./xc : exiting
memory management: default_pmm_manager
check_alloc_page() succeeded!
check_pgdir() succeeded!
check_boot_pgdir() succeeded!
----- BEGIN -----
PDE(e0) c0000000-f8000000 38000000 urw
  |-- PTE(38000) c0000000-f8000000 38000000 -rw
PDE(1) fac00000-fb000000 400000 -rw
  |-- PTE(e0) faf00000-fafe0000 e0000 urw
  |-- PTE(1) fafeb000-fafec000 1000 -rw
----- END -----
use SLOB allocator
check_slab() success
kmalloc_init() succeeded!
check_vma_struct() succeeded!
```

```
page fault at 0x5000: K/W [no page found].
swap_out: i 0, store page in vaddr 1000 to disk swap entry 2
write Virt Page b in fifo_check_swap
write Virt Page a in fifo_check_swap
PC : 0xc0007b7c

page fault at 0x1000: K/W [no page found].
do pgfault: ptep c05f3004, pte 200
swap_out: i 0, store page in vaddr 2000 to disk swap entry 3
SWAP: load ptep c05f3004 swap entry 2 to vaddr 0x1000, page c041cea0, No 1525
swap_in: load disk swap entry 2 with swap_page in vadr 0x1000
write Virt Page b in fifo_check_swap
PC : 0xc0007bb4

page fault at 0x2000: K/W [no page found].
do pgfault: ptep c05f3008, pte 300
swap_out: i 0, store page in vaddr 3000 to disk swap entry 4
SWAP: load ptep c05f3008 swap entry 3 to vaddr 0x2000, page c041cec0, No 1526
swap_in: load disk swap entry 3 with swap_page in vadr 0x2000
write Virt Page c in fifo_check_swap
PC : 0xc0007bec

page fault at 0x3000: K/W [no page found].
do pgfault: ptep c05f300c, pte 400
swap_out: i 0, store page in vaddr 4000 to disk swap entry 5
SWAP: load ptep c05f300c swap entry 4 to vaddr 0x3000, page c041cee0, No 1527
swap_in: load disk swap entry 4 with swap_page in vadr 0x3000
write Virt Page d in fifo_check_swap
PC : 0xc0007c24

page fault at 0x4000: K/W [no page found].
do pgfault: ptep c05f3010, pte 500
swap_out: i 0, store page in vaddr 5000 to disk swap entry 6
SWAP: load ptep c05f3010 swap entry 5 to vaddr 0x4000, page c041ce80, No 1524
swap_in: load disk swap entry 5 with swap_page in vadr 0x4000
write Virt Page e in fifo_check_swap
PC : 0xc0007c5c

page fault at 0x5000: K/W [no page found].
do pgfault: ptep c05f3014, pte 600
swap_out: i 0, store page in vaddr 1000 to disk swap entry 2
SWAP: load ptep c05f3014 swap entry 6 to vaddr 0x5000, page c041cea0, No 1525
swap_in: load disk swap entry 6 with swap_page in vadr 0x5000
write Virt Page a in fifo_check_swap
PC : 0xc0007c8c

page fault at 0x1000: K/R [no page found].
do pgfault: ptep c05f3004, pte 200
swap_out: i 0, store page in vaddr 2000 to disk swap entry 3
SWAP: load ptep c05f3004 swap entry 2 to vaddr 0x1000, page c041cec0, No 1526
swap_in: load disk swap entry 2 with swap_page in vadr 0x1000
count is 31245, total is 31245
count is 5, total is 5
check_swap() succeeded!
```

# 主要遇到的问题 – 编译器

---

- 编译器不能处理const变量作为函数参数，而且不报错
- 编译器不支持c99在函数中间定义变量
- 编译器不支持define，用GCC预处理器解决
- 编译器不支持struct内变量初始化，用初始化函数解决)
- 对于尚未解决的编译错误不能够报错，只能在编码时人工检查。



# 主要遇到的问题 – 仿真器

---

- 无TLB相关指令，在修改PDT后，TLB不会更新（原xv6中没有考虑到这个问题，在页机制启动前后是有问题的，目前的解决方案是，原内存操作后的TLB变为清空TLB。
- 仿真器在被零除后会直接挂掉而不是进入算术异常，暂未解决（不影响操作系统运行）。
- 仿真器没有完善调试功能，调试上比较头疼。



# 主要遇到的问题 – 内存管理

---

- ucore使用链接脚本，让内核代码在0xc0000000以上的空间，v9的编译器此时和物理内存一致，需要在启动页机制之后把程序中所有的指针+0xc0000000。
- 在启动页机制之后不能平滑过渡，需要手动汇编操作栈进行函数调用和返回。

# 主要遇到的问题 – 进程控制

---

- 系统调用需要使用寄存器传参数，其中SYS\_exec有4个参数，v9只有3个寄存器
  - 需要使用类似args[]数组作为一个参数
- 进程控制中有和硬件密切相关的内容，包括trapframe和context，ucore和xv6在实现上有一些差异
  - 需要阅读ucore和xv6的代码和查阅相关理论，使用汇编实现

# 后续计划

---

- 文件系统调试完成，细节文档。

谢谢