



國立陽明交通大學

NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Institute of Artificial Intelligence Innovation

Department of Computer Science

Operating System

Homework 02: Memory Management

Shuo-Han Chen (陳碩漢),

shch@nycu.edu.tw

Thur. 13:20 - 16:20 ED305

Goal

1. Understand how memory management works in NachOS
2. Understand how to implement page table mechanism

Part 1 Trace Code

- Starting from “`threads/kernel.cc Kernel::ExecAll()`”, “`threads/thread.cc Thread::Sleep()`” until “`machine/mipssim.cc Machine::Run()`” is called for executing the first instruction from the user program.
- You need to explain at least the function mentioned below in the report.
 1. `threads/thread.cc`
 - `Thread::Sleep()`, `Thread::StackAllocate()`, `Thread::Finish()`, `Thread::Fork()`
 2. `userprog/addrspace.cc`
 - `AddrSpace::AddrSpace()`, `AddrSpace::Execute()`, `AddrSpace::Load()`
 3. `threads/kernel.cc`
 - `Kernel::Kernel()`, `Kernel::ExecAll()`, `Kernel::Exec()`, `Kernel::ForkExecute()`
 4. `threads/scheduler.cc`
 - `Scheduler::ReadyToRun()`, `Scheduler::Run()`

Part 1 Trace Code

- Answer the following questions in your report
 1. Explain how NachOS creates a thread (process), load it into memory and place it into the scheduling queue.
 2. How does Nachos allocate the memory space for a new thread(process)?
 3. How does Nachos initialize the memory content of a thread(process), including loading the user binary code in the memory?
 4. How does Nachos create and manage the page table?
 5. How does Nachos translate addresses?
 6. How Nachos initializes the machine status (register, etc) before running a thread (process)
 7. Which object in Nachos acts the role of process control block

Part 2 Implementation

- Modify its memory management code to make NachOS support multiprogramming.
- Without multi-programming

```
../build.linux/nachos -e consoleIO_test1 -e consoleIO_test2
consoleIO_test1
consoleIO_test2
9
16
15
18
19
1return value:0
7
return value:0
```

- After **implement multi-programming**

```
../build.linux/nachos -e consoleIO_test1 -e consoleIO_test2
consoleIO_test1
consoleIO_test2
9
8
7
6
1return value:0
5
16
17
18
19
return value:0
```

Part 2 Prerequisite

- Before you start implement this homework feature, you need to ensure your previous NachOS assignments meet the following requirements.
 1. ConsoleIO implementation
 2. consoleIO_test file modification
 3. test scripts

ConsoleIO Implementation

- If you don't follow the correct method for printing an integer, the result will also be incorrect. The following code demonstrates one of the proper ways to do it.

- exception.cc

```
case SC_PrintInt:
    val = kernel->machine->ReadRegister(4);
    SysPrintInt(val);
    kernel->machine->WriteRegister(PrevPCReg, kernel->machine->ReadRegister(PCReg));
    kernel->machine->WriteRegister(PCReg, kernel->machine->ReadRegister(PCReg) + 4);
    kernel->machine->WriteRegister(NextPCReg, kernel->machine->ReadRegister(PCReg)+4);
    return;
```

- ksyscall.h

```
void SysPrintInt(int value)
{
    kernel->interrupt->PrintInt(value);
}
```

- interrupt.cc

```
void Interrupt::PrintInt(int value)
{
    kernel->synchConsoleOut->PutInt(value);
}
```

- synchconsole.cc

```
void
SynchConsoleOutput::PutInt(int value)
{
    char str[15];
    int idx=0;
    sprintf(str, "%d\0", value);
    lock->Acquire();
    while (str[idx] != '\0') {
        consoleOutput->PutChar(str[idx]);
        idx++;
        waitFor->P();
        kernel->stats->numConsoleCharsWritten--;
    }
    consoleOutput->PutChar('\n');
    waitFor->P();
    lock->Release();
}
```

ConsoleIO_test modification

- consoleIO_test1.c before consoleIO_test1.c after

```
#include "syscall.h"
int
main()
{
    int n;
    for (n=9;n>5;n--) {
        PrintInt(n);
    }
    Halt();
}
```

```
#include "syscall.h"
int
main()
{
    int n;
    for (n=9;n>5;n--) {
        PrintInt(n);
    }
    // Halt();
    return 0;
}
```

- consoleIO_test2.c before consoleIO_test2.c after

```
#include "syscall.h"
int
main()
{
    int n;
    for (n=15;n<=19;n++){
        PrintInt(n);
    }
    Halt();
}
```

```
#include "syscall.h"
int
main()
{
    int n;
    for (n=15;n<=19;n++){
        PrintInt(n);
    }
    // Halt();
    return 0;
}
```


Test Scripts

- Since we do not use the Halt command to stop NachOS (as we want to avoid shutting down NachOS after the completion of each user program).
- The NachOS program will not shut down by itself, even after finishing multiple tasks.
- Therefore, in your test scripts, you will need to modify your commands as follows:

```
make distclean
make
timeout 1 ../build.linux/nachos -e consoleIO_test1 -e consoleIO_test2
echo "done"
```

Requirements

- Implement a data structure to record used physical memory.
- Allocate the physical memory correctly.
- When the thread is finished, make sure to release the address space and restore physical page status.

Hint

- The following files “may” be modified...
 - userprog/addrspace.*
 - threads/kernel.*

Jenkins verification

- The TA's job will involve running two tests.
 1. The first one uses the following command:
“`../build.linux/nachos -e consoleIO_test1 -e consoleIO_test2`”
 2. The second is a hidden test, but you will need to ensure that your output contains all the expected values."

```
20
22
9
8
24
26
28
15
16
17
7
6
1return value:0
8
19
return value:0
30
32
34
36
38
40
return value:0
```

Grading

- Part1 (Trace) - 38%
 1. Explain function - 19%
 2. Answer question - 19%
- Part2 (Implementation)- 60%
 1. Each test case - 30%
- Report Format - 2%
- Deadline: 11/30

Report Format

- Please follow the word file to form your report for HW02
- Format guide
 - Content format: should be set with 12pt front, 16pt row height, and align to the left.
 - Caption format: 18pt and Bold font.
 - Font format: Times New Roman, 標楷體
 - Figure: center with single line row height.
 - Change the title to your student ID and name in Chinese.
 - Upload pdf file with the file name format : OS_HW02_GROUP_X.pdf (change X to your group ID)

Reminder

- 0 will given to cheaters. Do not copy & paste!
 - TA will check your repository
- Feel free to ask TA questions
 - Teams Message(Recommended): 廖永誠
 - Email: yongchengliaw.ii12@nycu.edu.tw

Q & A

Thank you for your attention