CPU Virtualization

- OS
- Virtual, Concurrency, Persistence

CPU

Mem.

- Process - machine state $\Rightarrow$ PCB.

$\text{PCB}(P_1)$

$P_1$

$\text{PCB}(P_2)$

$P_2$

- Program vs Process

Program

- data
- code

Process

- stack
- heap
- data
- code

CPU: $P_1, P_1, P_2, P_2$

$0 \ 1 \ 2 \ 3 \ 4 \ t$
Scheduling

Mechanisms

1. LDE
2. Context Switch
3. Policies

Scheduler - component of OS.

LDE

Direct Execution.

Q: How can we use the OS efficiently virtualize the CPU?

OS P1 OS P2 OS P1...
Prob #1: Restricted Ops

Soln: Kernel mode vs user mode

System Calls

0 - user
1 - kernel

C

printf(...)

write(...)

TRAP

OS

write(...) {
    yield();
}

RETURN FROM TRAP

Kernel level

User level

Prob #2: How to switch from Pi to OS?

Soln: Timer interrupt
System call

write(…)

Assembly:

int $64

\texttt{\textbackslash r eax 3}

TRAP

OS

Trap Handler

syscall ( ) {
  write();
}

Return from Trap

X86 - Hlw

Xv6 - OS

open
read
write

1

2

3
Example: Linux "write" system call

C code:
```c
...  
printf("Hello world\n");  
...  
```

libc:
```
%eax = sys_write;  
int 0x80  
```

```c
system_call()
{
    sc = sys_call_table[%eax]
}
```

```c
sys_write(...)
{  
    //handle write
}
```
HW → user level → Trap Frame

OS → kernel level → Context

%esp → context → TF → kstack

K → kstack
Context Switch (From Proc A to scheduler)

Process A's Kernel Stack

- trapframe
- %esp
- %eip (all trap)
- trap locals
- %eip (trap)
- yield locals
- %eip (yield)
- sched locals
- cpu → scheduler
- &proc → context
- %eip (sched)
- %ebp
- %ebx
- %esi
- %edi

Scheduler Stack

- &proc → context
- cpu → scheduler
- %eip (scheduler)
- %ebp
- %ebx
- %esi
- %edi
- %eax
- %edx
- %esp

NOTE: Arrows (→) indicate the state after switching from Proc A to scheduler.
Context Switch
From scheduler to Proc B

Scheduler Stack

Process B's Kernel Stack

trapframe
- %esp
- %eip (all traps)
- trap locals
- %eip (trap)
- yield locals
- %eip (yield)
- sched locals
- cpu -> scheduler
- &proc -> context
- %eip (sched)
- %ebp
- %ebx
- %esi
- %edi

NOTE: Arrow indicate the state after switching from scheduler to B.