

EEE3052: Introduction to Operating Systems

Fall 2017

Project #1

Project Plan

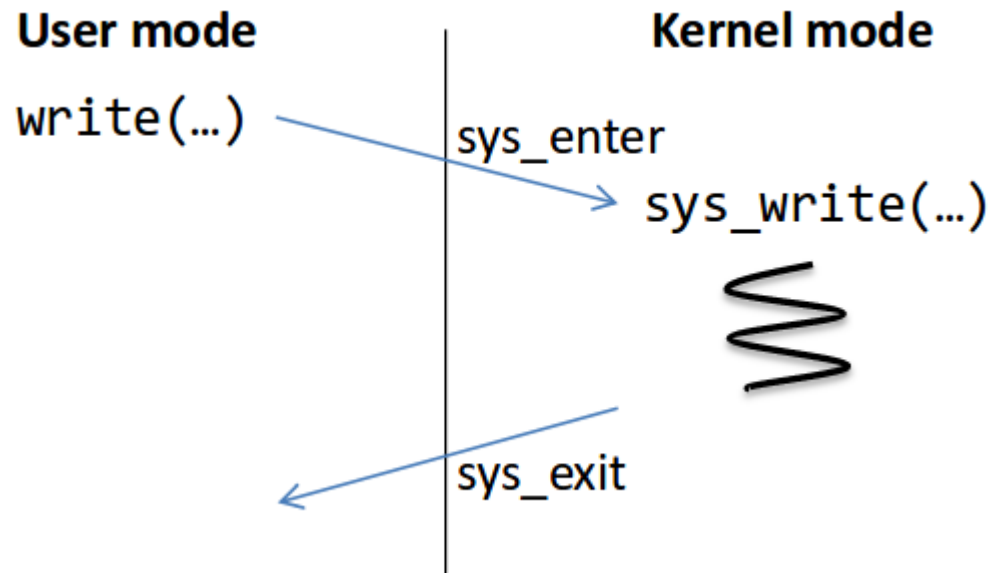
- 4 projects
 - 0) Install Xv6
 - 1) Process management
 - System call (9/11 ~ 9/17)
 - Scheduling
 - 2) Virtual memory
 - 3) Synchronization
 - 4) File system

Process

- Program: executable code
- Process
 - A running instance of program
 - Runtime resources
 1. CPU: a set of registers
 2. Memory: code, data, heap, stack, etc.
 3. File: open files, sockets, device handle, etc.
 - User mode & Kernel mode

System Call

- An extended function call
 - From applications in user space
 - To operating system functions
 - Access system resources safely



System Call on Xv6

- User process P can only see its own memory
 - Because it is user mode
- Other areas such as kernel are hidden



System Call on Xv6 (Cont.)

- Process P calls *fork()* system call

```
movl    $1, %eax
int     $64
```



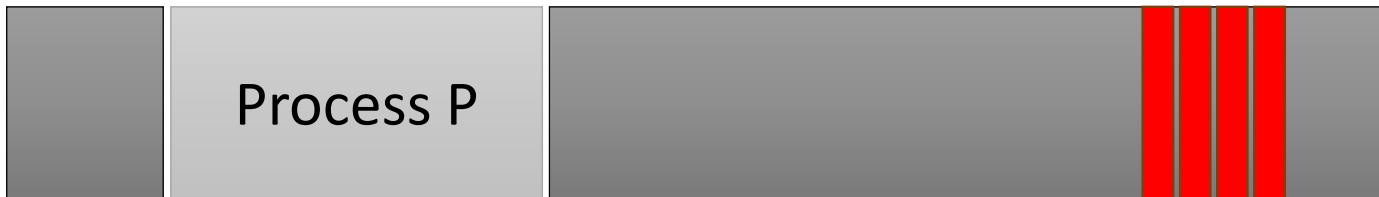
System Call on Xv6 (Cont.)

- Process P calls *fork()* system call

system call
table index

```
movl    $1, %eax  
int     $64
```

Physical
Memory



System Call on Xv6 (Cont.)

- Process P calls *fork()* system call

```
movl    $1, %eax
int     $64
```

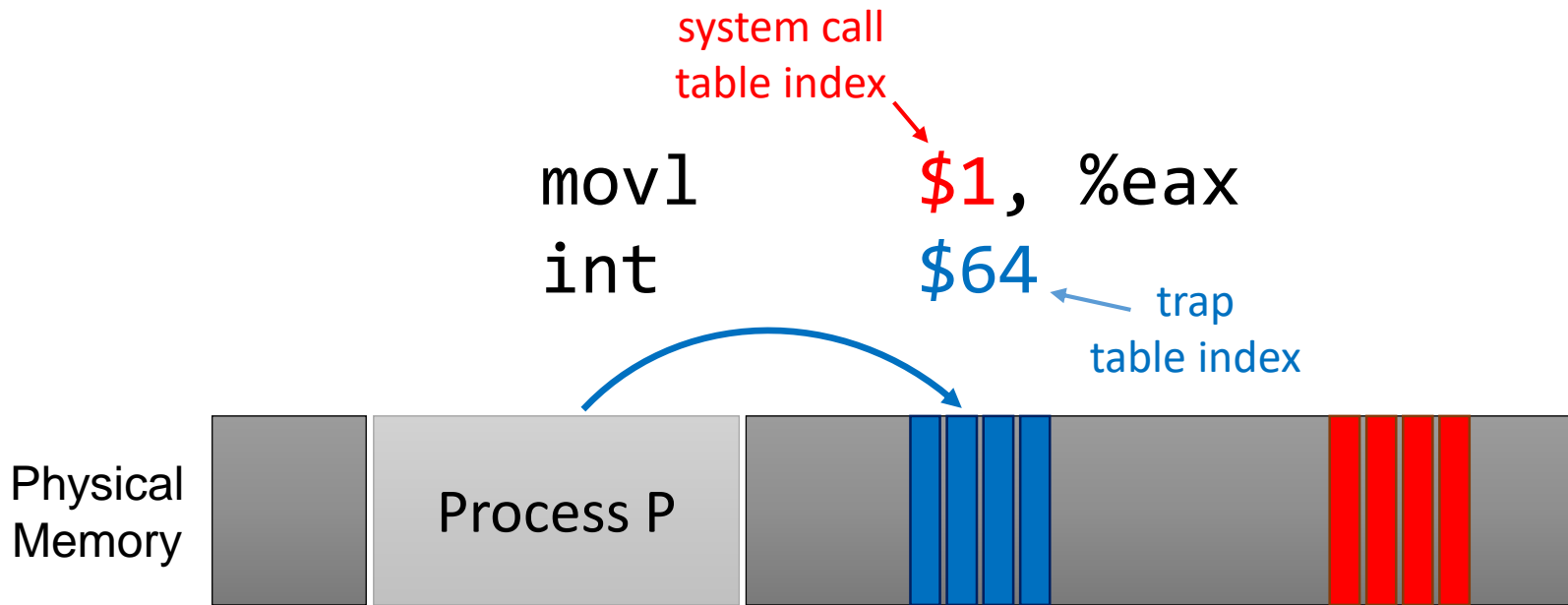
system call table index

trap table index



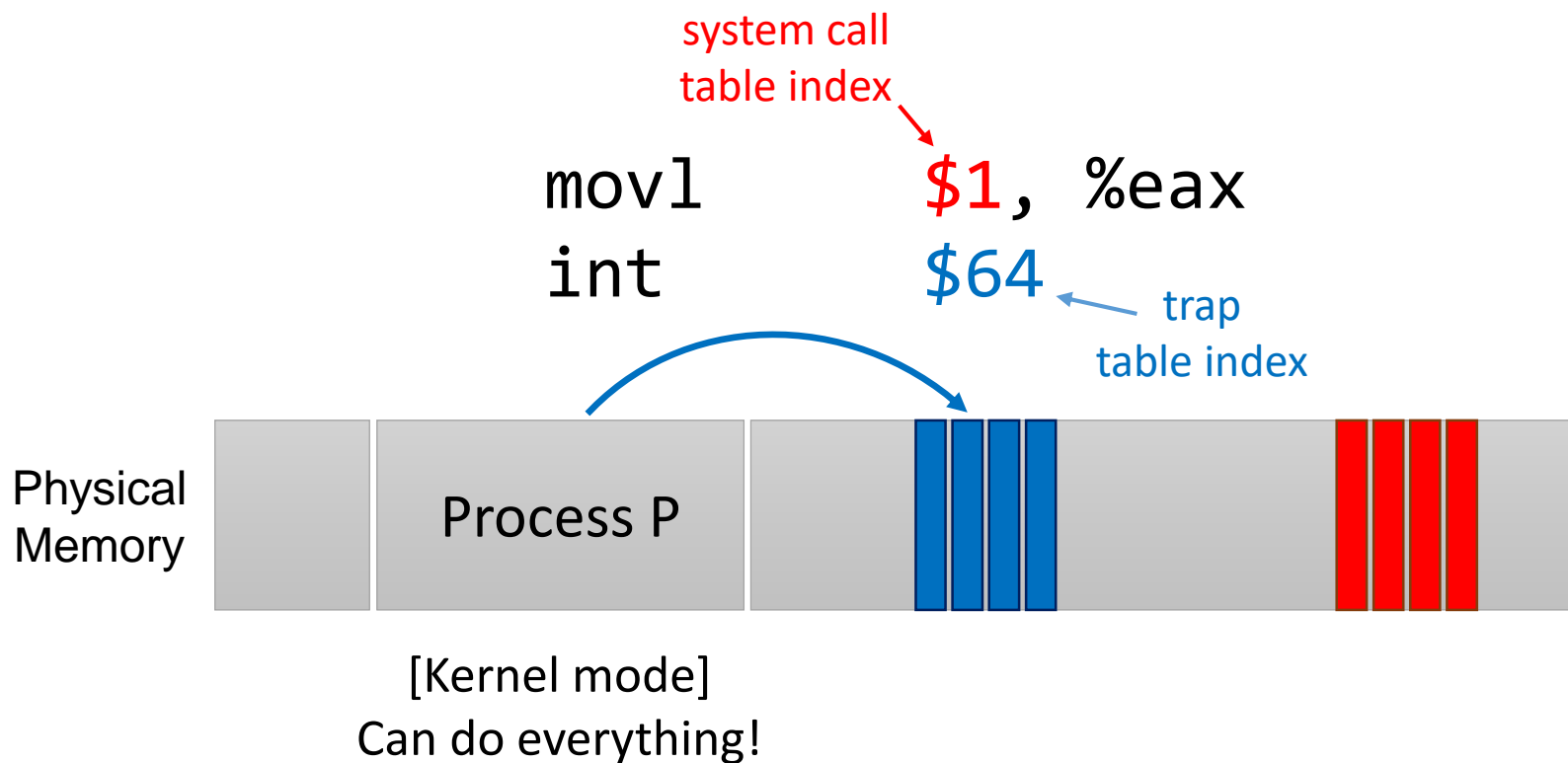
System Call on Xv6 (Cont.)

- Process P calls *fork()* system call



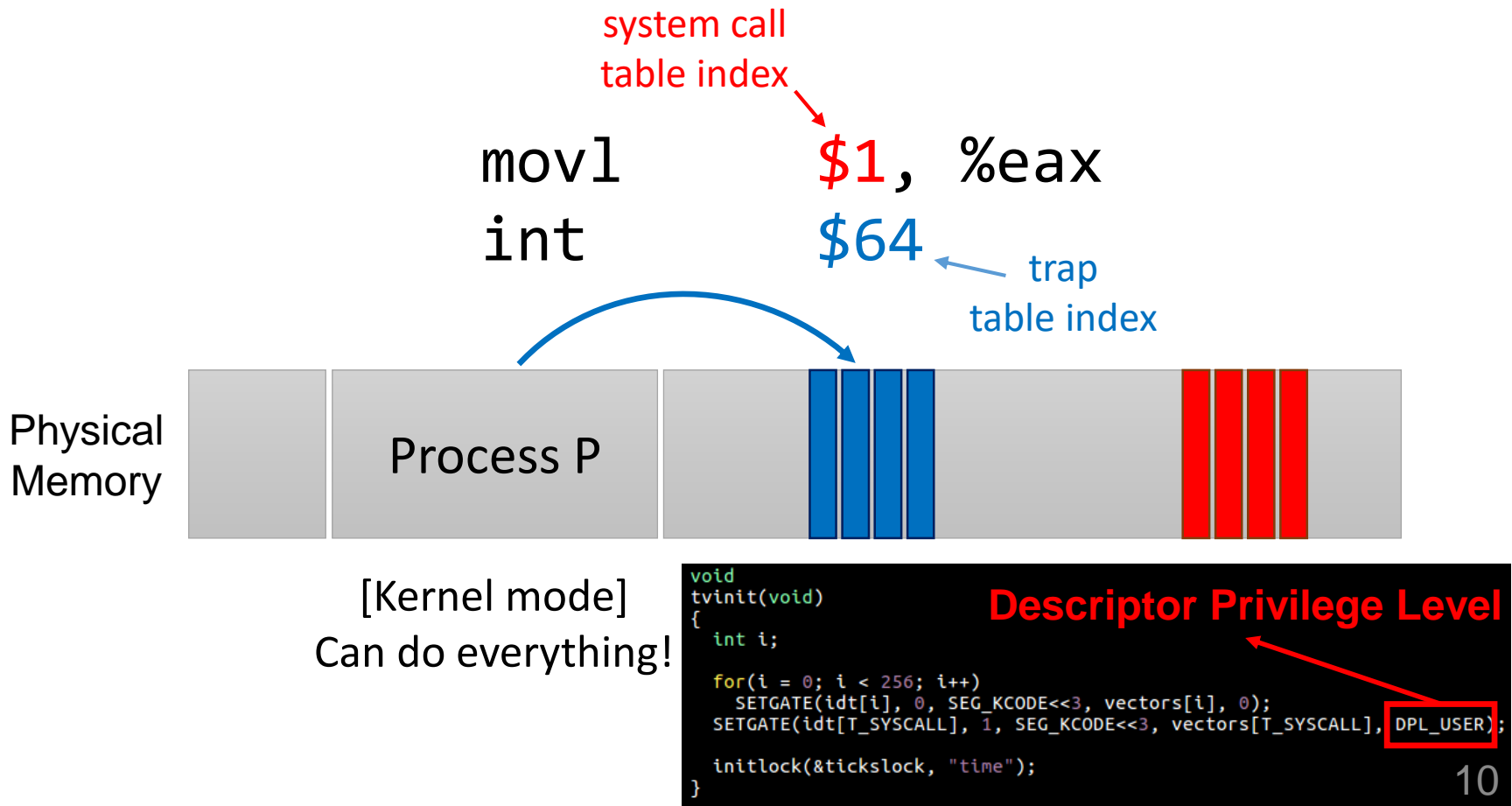
System Call on Xv6 (Cont.)

- Process P calls *fork()* system call



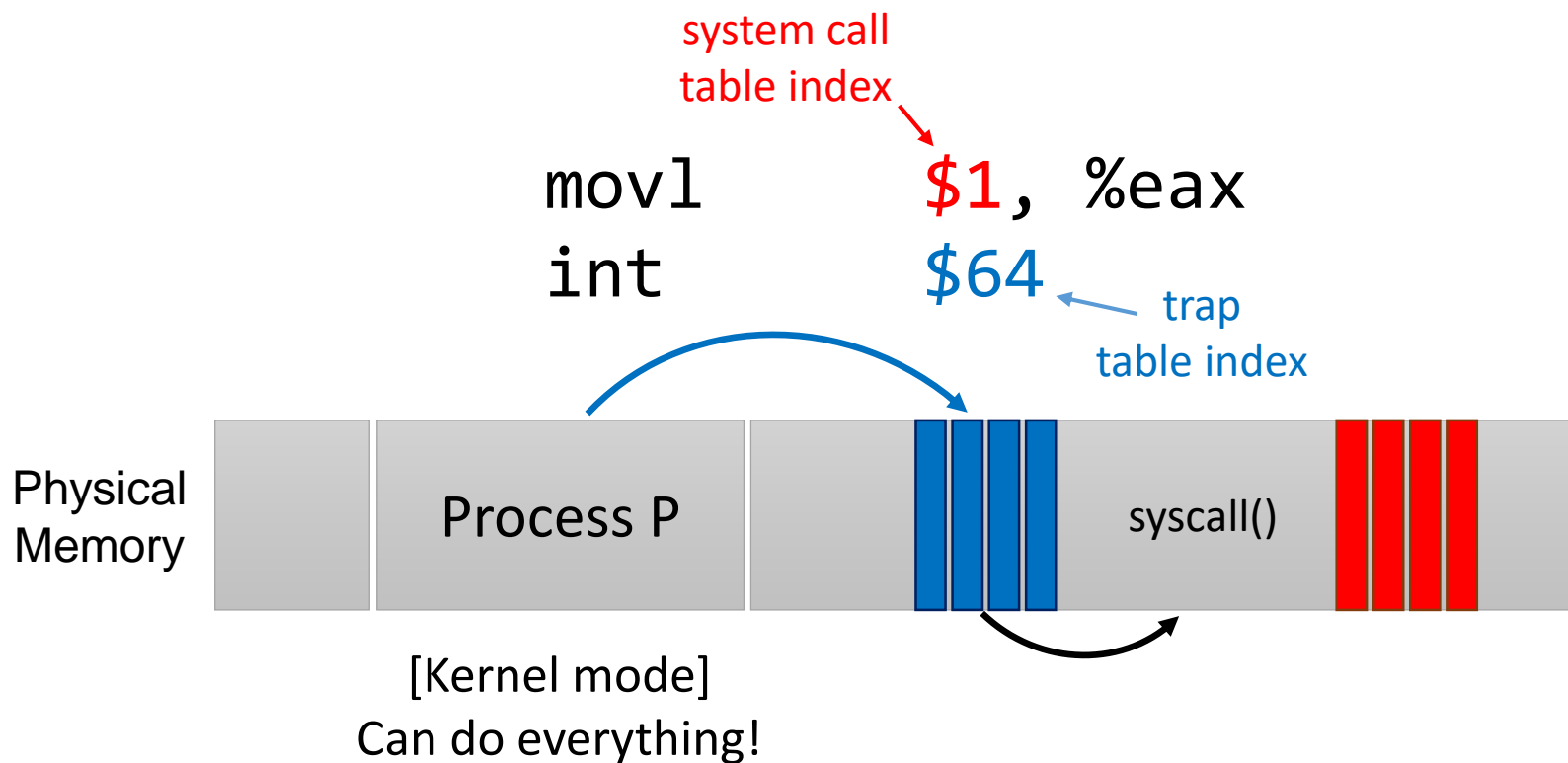
System Call on Xv6 (Cont.)

- Process P calls *fork()* system call



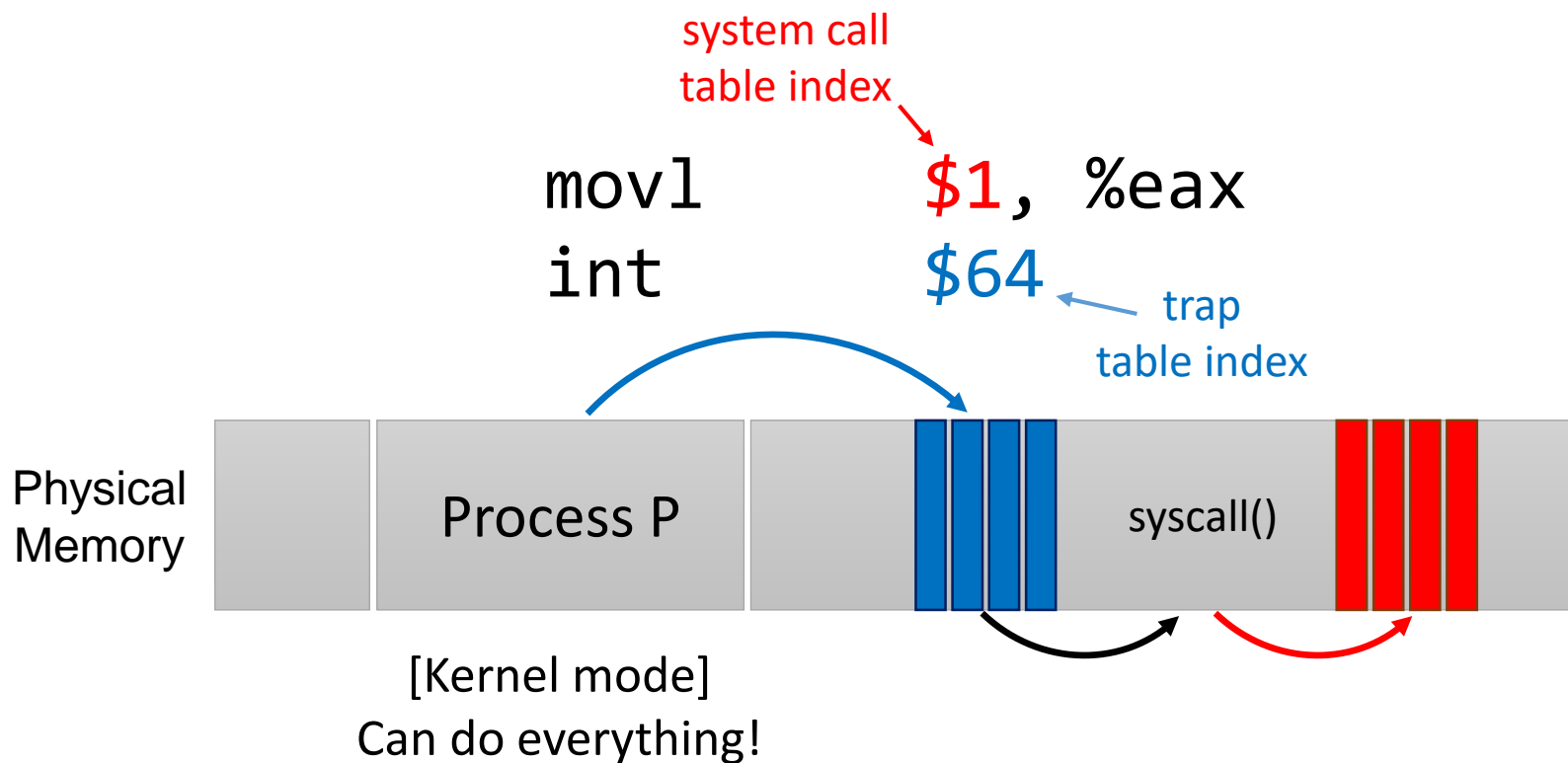
System Call on Xv6 (Cont.)

- Process P calls *fork()* system call



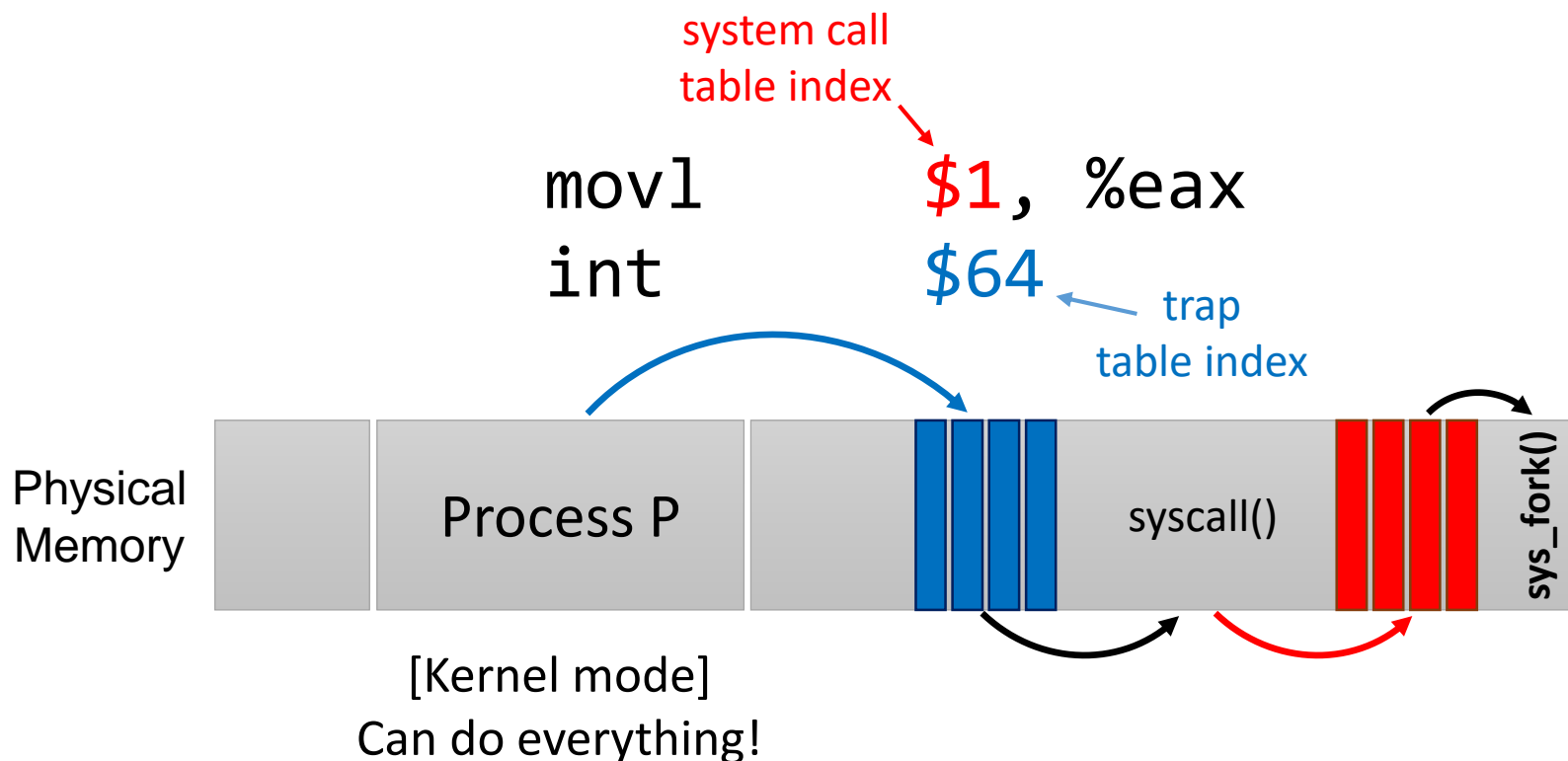
System Call on Xv6 (Cont.)

- Process P calls *fork()* system call



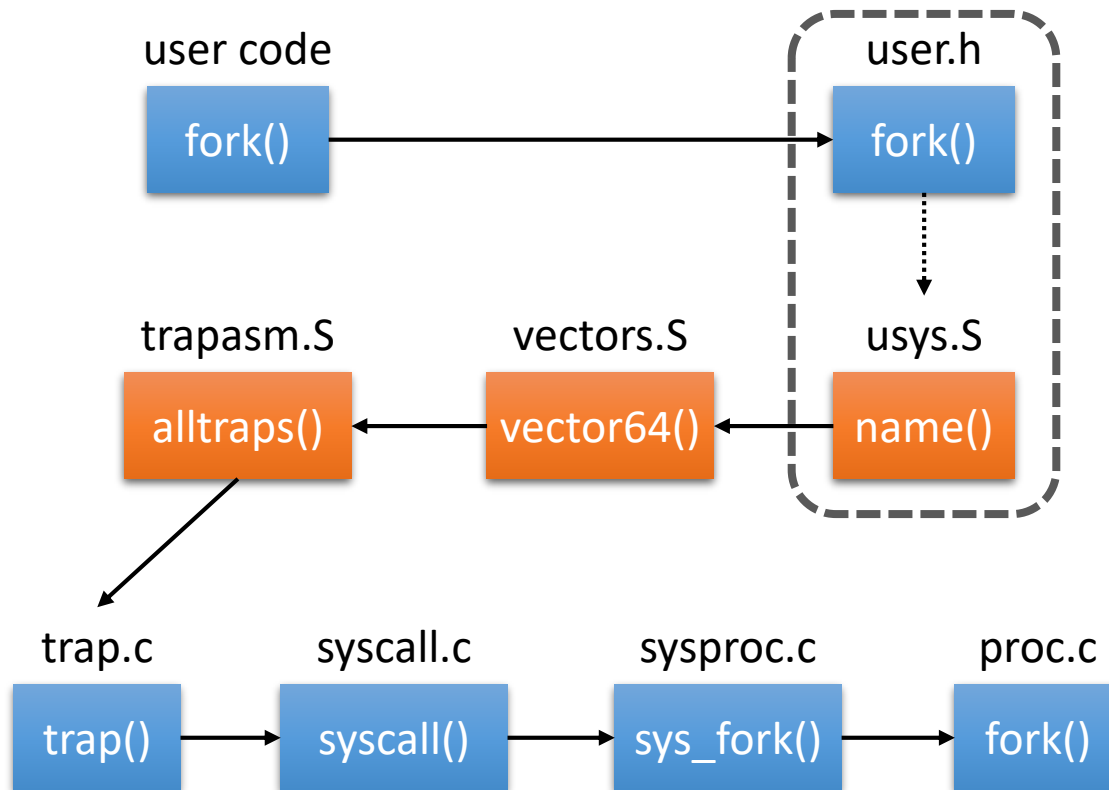
System Call on Xv6 (Cont.)

- Process P calls *fork()* system call



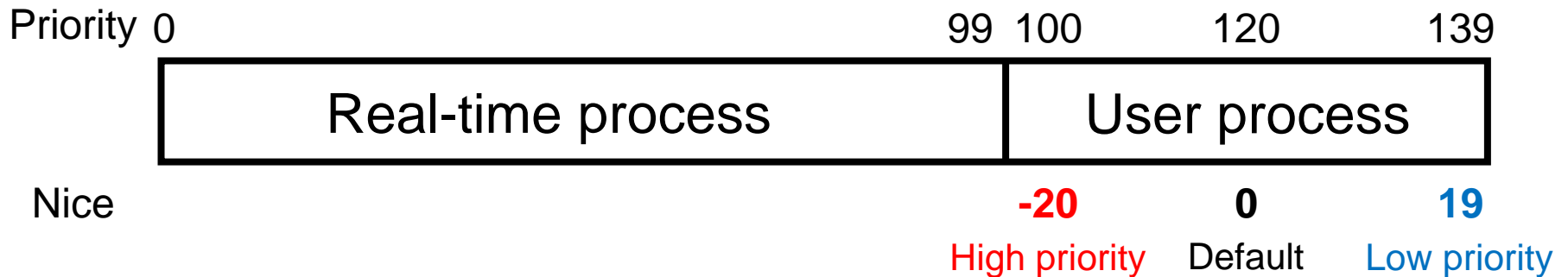
System Call Flow on Xv6

- *fork* system call



Process Priority

- For represent weight between processes
- In Linux



- In Xv6
 - No implementation!

Set Process Priority

- User process set priority by using nice value
 - setnice
 - Set process priority
 - Argument: process ID, nice value
 - Return value
 - If set priority failed, return -1
 - Else, return 0
 - getnice
 - Return process priority
 - Argument: process ID
 - Return value
 - If process corresponding ID does not exist, return -1
 - Else, return process priority

Linux Command - *top*

- Process monitoring command

```
top - 22:40:38 up 5 days, 1:21, 1 user, load average: 0.84, 1.21, 1.36
Tasks: 204 total, 1 running, 203 sleeping, 0 stopped, 0 zombie
%Cpu(s): 11.1 us, 8.3 sy, 0.2 ni, 80.2 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem : 16310032 total, 1257964 free, 11307036 used, 3745032 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 3088832 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2416	jaeh	20	0	4557772	1.810g	220732	S	40.0	11.6	614:28.53	firefox
21226	jaeh	20	0	11.230g	8.424g	8.284g	S	37.3	54.2	108:05.55	VirtualBox
2193	jaeh	9	-11	437252	16492	10064	S	3.7	0.1	42:52.01	pulseaudio
2220	jaeh	20	0	1448020	280988	76084	S	1.3	1.7	146:52.50	compiz
21191	jaeh	20	0	962144	24384	18360	S	0.7	0.1	0:36.86	VBoxSVC
937	root	20	0	167720	8912	8188	S	0.3	0.1	0:36.55	thermald
1103	root	20	0	1768660	503696	473036	S	0.3	3.1	78:46.84	Xorg
21166	jaeh	20	0	1542868	113908	86740	S	0.3	0.7	0:28.14	VirtualBox
21186	jaeh	20	0	167880	12628	10004	S	0.3	0.1	0:16.24	VBoxXPCOMIPCD
22237	jaeh	20	0	618252	47584	35236	S	0.3	0.3	0:00.41	gnome-terminal-
1	root	20	0	120036	6068	3864	S	0.0	0.0	0:09.59	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.14	kthreadd

Project #1-1 – System call

- Make two system calls
 - int setnice(int pid, int nice_value)
 - int getnice (int pid)
- Make process status monitoring - *minitop*
 - Print all process status
 - Contents
 - 1) Process ID
 - 2) Parent process ID
 - 3) Process priority
 - 4) Process state
 - 5) Process name

EMBRYO
SLEEPING
RUNNABLE
RUNNING
ZOMBIE



```
csl@simul:~/xv6-public$ make qemu-nox
qemu-system-i386 -nographic -drive file=
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nl
init: starting sh
$ minitop
pid      ppid     prio     state    name
1         1        20      sleep   init
2         1        20      sleep   sh
3         2        20      run     minitop
$
```

Project #1-1 Score

- Total 100 point
- Checkpoint
 - Get nice of init process (must 20) – 10p
 - Get nice of non-existing process (wrong pid) – 10p
 - Set nice to current process – 10p
 - Set nice to non-existing process (wrong pid) – 10p
 - Set wrong nice on current process – 20p
 - Get nice of forked process – 20p
 - Forked process inherits parent process priority
 - Print all process status (minitop) – 20p

Submit

- Send email to jaehyun.song@csl.skku.edu
 - Title: [EEE3052]Project-1_1-studentID-name
 - File name: studentID-1_1.tar.gz

- Due date
 - 2017/09/17 Sunday 23:59:59
 - Penalty **10%** of each project score per **one day**

- TA contact
 - jaehyun.song@csl.skku.edu **(ONLY THIS EMAIL)**

Compress Xv6-public

- cd
- tar cvf studentID-1_1.tar.gz xv6-public
 - Example: tar cvf 2013456789-1_1.tar.gz xv6-public

```
csl@simul:~$ tar cvf 2013456789-1_1.tar.gz xv6-public
```