OSTEP Chapter 6

ECE 3600, Fall 2022

Table of Contents

- <u>1. Direct Execution</u>
- 2. Limited Direct Execution
- 3. Limited Direct Execution with Timer

1. Direct Execution

user mode vs. kernel mode, system call, trap, timer interrupt

scheduler, context switch

OS	Program
Create entry for process list	
Allocate memory for program	
Load program into memory	
Set up stack with argc/argv	
Clear registers	
Execute call main()	
()	Run main()
	Execute return from main
Free memory of process	

Remove from process list

Figure 6.1: Direct Execution Protocol (Without Limits)

2. Limited Direct Execution

OS @ boot (kernel mode)	Hardware	
initialize trap table	remember address of syscall handler	
OS @ run (kernel mode)	Hardware	Program (user mode)
Create entry for process list Allocate memory for program Load program into memory Setup user stack with argv Fill kernel stack with reg/PC return-from-trap		
	restore regs (from kernel stack) move to user mode jump to main	
) 1	Run main()
		 Call system call trap into OS
Handla tran	save regs (to kernel stack) move to kernel mode jump to trap handler	_
Handle trap Do work of syscall return-from-trap		
	restore regs (from kernel stack) move to user mode jump to PC after trap	
Free memory of process		 return from main trap (via exit ()
Remove from process list		

Figure 6.2: Limited Direct Execution Protocol

OS @ boot (kernel mode)	Hardware	
initialize trap table start interrupt timer	remember addresses of syscall handler timer handler start timer interrupt CPU in X ms	
OS @ run (kernel mode)	Hardware	Program (user mode) Process A
Handle the trap Call switch() routine save regs(A) \rightarrow proc_t(A)	timer interrupt save regs(A) → k-stack(A) move to kernel mode jump to trap handler	
restore regs(B) ← proc_t(B) switch to k-stack(B) return-from-trap (into B)	restore regs(B) \leftarrow k-stack(B) move to user mode jump to B's PC	Process B

3. Limited Direct Execution with Timer

Figure 6.3: Limited Direct Execution Protocol (Timer Interrupt)