CS3210: The Weird Machine

Tutorial

Agenda

- Understanding what is weird machine and a demo.
- In class exercise:
  - Implementing the page directory and table information in JOS

The Weird Machine

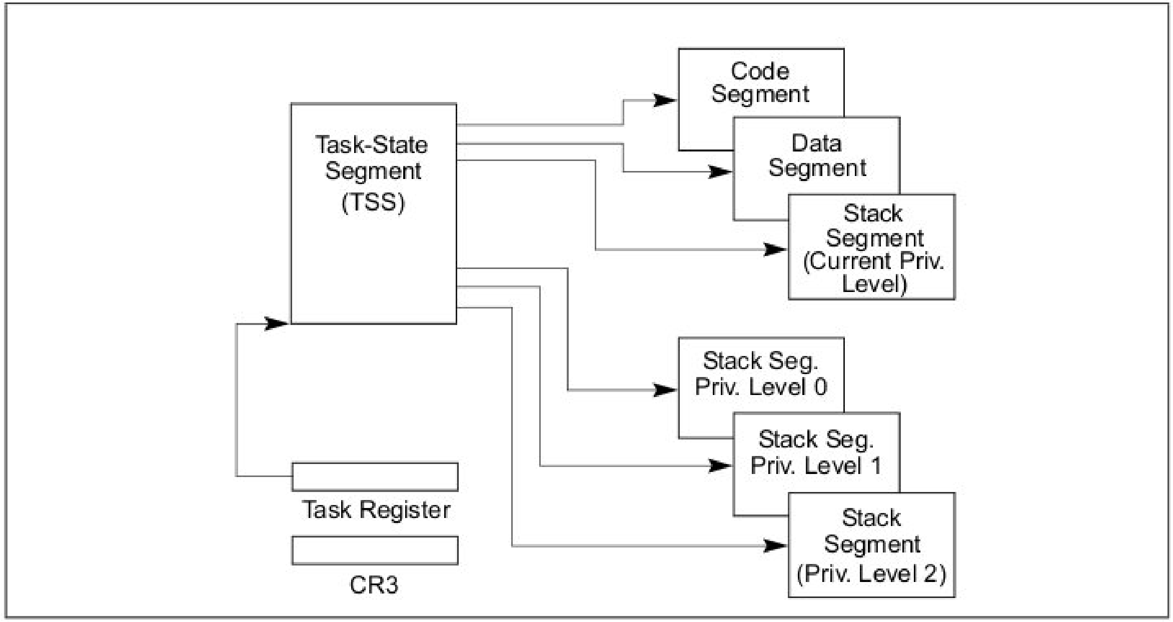
Q. What do you understand from "The Weird Machine"?

- Type of abstracted programming model comprising of undefined or arbitrary behaviors
- Additional code execution outside specification of a program
- Examples: format strings exploits, heap overflow, undefined OS traps

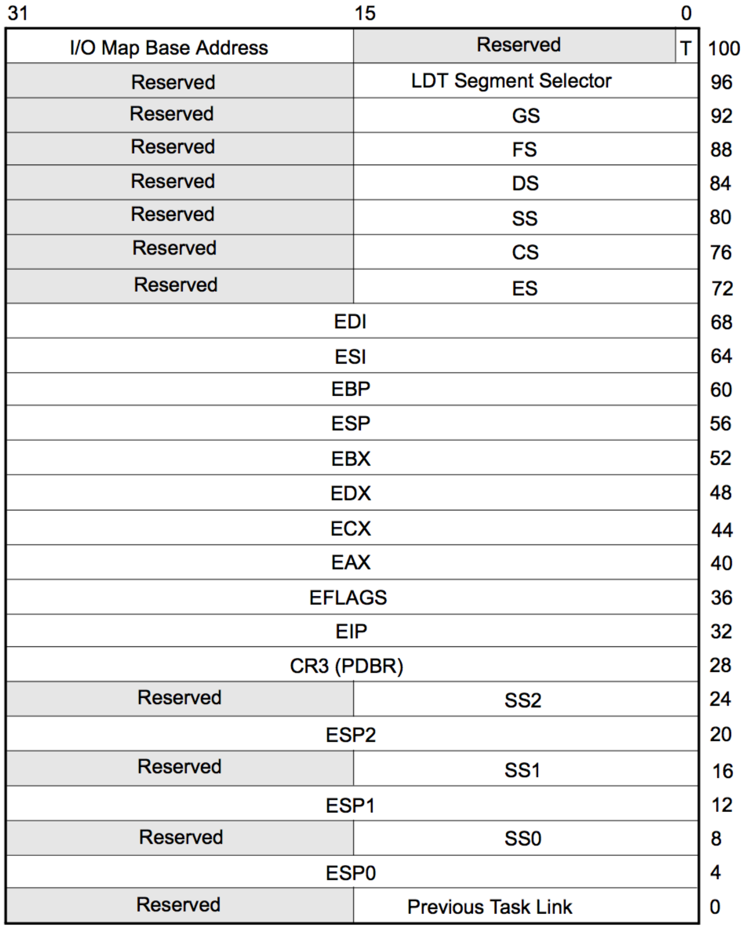
Example

- Series of faults and double faults without executing any instruction
- Relies on interrupt handling (GDT/IDT) and memory translation handling

Task State Segment

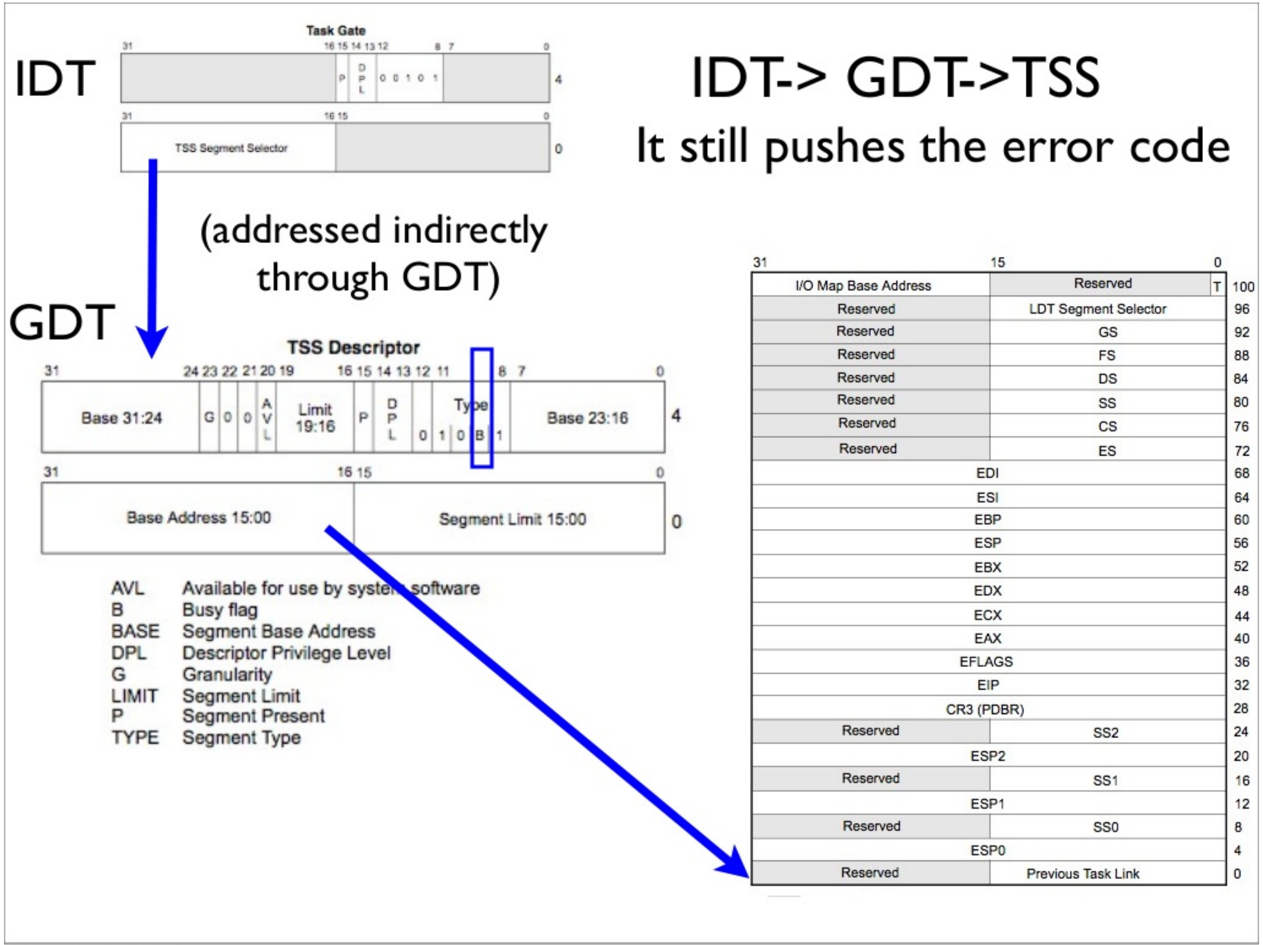


Task State Segment



31	15	0	
I/O Map Base Address	Reserved	T	100
Reserved	LDT Segment Selector		96
Reserved	GS		92
Reserved	FS		88
Reserved	DS		84
Reserved	SS		80
Reserved	CS		76
Reserved	ES		72
EDI			68
ESI			64
EBP			60
ESP			56
EBX			52
EDX			48
ECX			44
EAX			40
EFLAGS			36
EIP			32
CR3 (PDBR)			28
Reserved	SS2		24
ESP2			20
Reserved	SS1		16
ESP1			12
Reserved	SS0		8
ESP0			4
Reserved	Previous Task Link		0

Page-fault Handling



IDT-> GDT->TSS

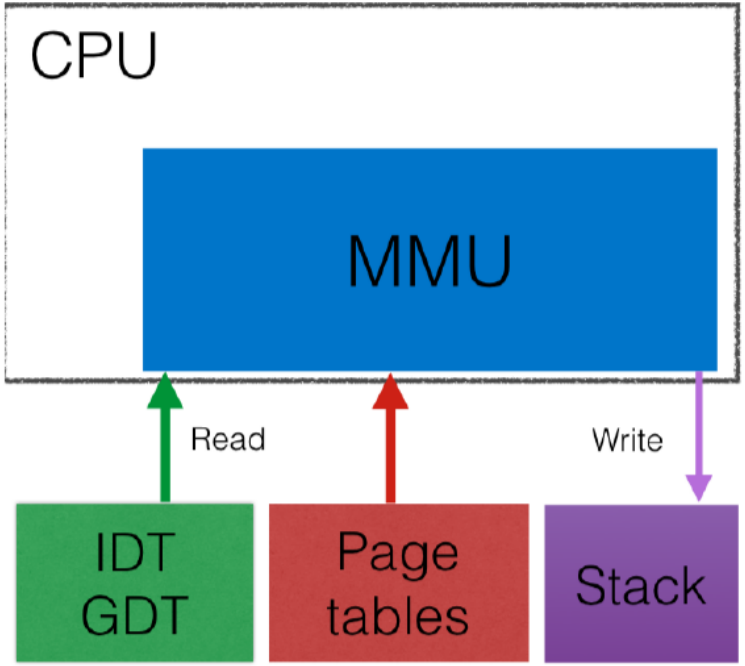
It still pushes the error code

(addressed indirectly through GDT)

AVL	Available for use by system software
B	Busy flag
BASE	Segment Base Address
DPL	Descriptor Privilege Level
G	Granularity
LIMIT	Segment Limit
P	Segment Present
TYPE	Segment Type

31	15	0	
I/O Map Base Address	Reserved	T	100
Reserved	LDT Segment Selector		96
Reserved	GS		92
Reserved	FS		88
Reserved	DS		84
Reserved	SS		80
Reserved	CS		76
Reserved	ES		72
EDI			68
ESI			64
EBP			60
ESP			56
EBX			52
EDX			48
ECX			44
EAX			40
EFLAGS			36
EIP			32
CR3 (PDBR)			28
Reserved	SS2		24
ESP2			20
Reserved	SS1		16
ESP1			12
Reserved	SS0		8
ESP0			4
Reserved	Previous Task Link		0

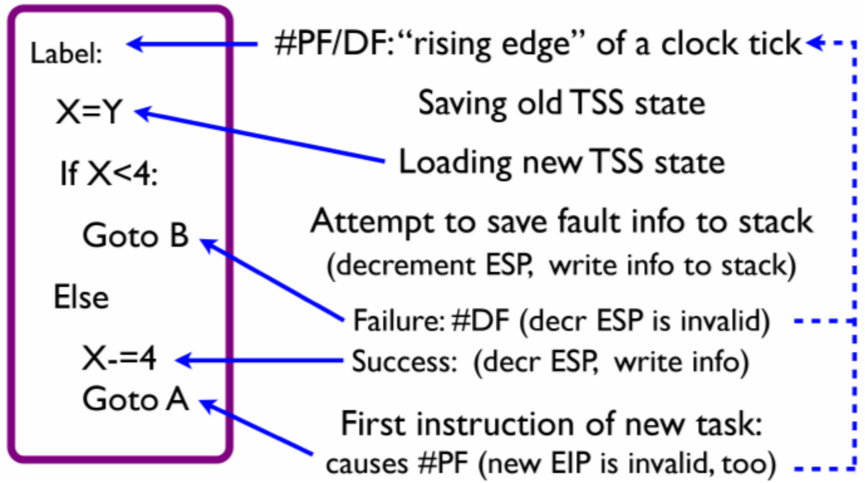
High Level Idea



- What if the stack address is not valid?

Approach

- Uses Turing complete *movdbz* instruction
  - move-branch-if-zero-or-decrement instruction


Label:
X=Y
If X<4:
Goto B
Else
X-=4
Goto A
#PF/DF:"rising edge" of a clock tick
Saving old TSS state
Loading new TSS state
Attempt to save fault info to stack
(decrement ESP, write info to stack)
Failure: #DF (decr ESP is invalid)
Success: (decr ESP, write info)
First instruction of new task:
causes #PF (new EIP is invalid, too)

Demo

Tutorial

```
$ git clone git://tc.gtisc.gatech.edu/cs3210-pub
```

or

```
$ cd cs3210-pub
$ git pull
```