S2 bit or more depends on what mode you are operating

64bit twice as many registors

2^48 physical address change,

Registors: More FP

Control registors: CR(special things)

00:Paging off protected mode

Cr3: pointers to registors

Debug: address from ip(debugger break point)

MSR:

MMU: Real,Protected

Turn off the processors, the registors are set to known

Bootloader:

The role of BIOS is to get piexes of boot sequence loader

Boot sector from the defined boot

BIOS first loads from hardware

Sector number of boot sector Second stage boot loader:

Windows (tfs) linux: XFS

The second stage boot loader to kernel(not reaaly in real mode may be an agreement between os bbot loader and kernel)

Might still be running in real mode

Loading fro CRO

:movl

a->b

orl ] (turn bits on all =bits

pe(PROTECTION AND PAGE MODE)

first process: UIX init

(it depends on the OS)

EVERY VERSION: 4 PREVILEGE LEVELS(numeric 0-3)

Most privilege level:0;(os kernel, )

Other not used

(OS 2 : it uses ather privilege levels-30 years ago)pedlars

Segmented addressing:

F000:FFF0 IN 64 IT (IGNORE INTIAL ONES)0-FFFFF ONE SEGMENT

Interrupt: Bunch of devices in any machine, it is a request from software are hardware:

Hard disk, usb, keyboard, firewall, graphic card, battery,(interrupts through a configuration events)

Tat -Z TOTAL NUMBER OF INTERRURPTS: RATE 399 /SEC4 cpus so 400 interrupts:

Each of my cpus is getting interrupt s 100 per sec

Ipi: Inter process interrupt: 112/sec for 4

Acpi interrupt: Advanced conf

Inteldrmo☹graphics card): at this memory location(no. of commands to do after wards interrupt)

Xhci: USB 3

Ehci: usb slote

Azalia( sound card)

Pops skips( sometimes buffering)

Sdhc: ssd card slot)

Ppbl: bus

Iwn: wireless sending signals

Ehcil:

Ssd harddisk

pckb0:

2 keys 6 interrupts

200 interrupts per cpu:

The numbers IDT

InterruEACH BLOCK CONTAINS AN ADDRESS pt description table 0 -255

Looks into the table, puts the ciurrent state into stacks and handle the events in the event handler,

The longer in the event handler, less time,

Interrupt handlers should bre fast\

Disk controller to read a lotof data: priorities are present

Hypervisor: 32 bit 4gb

Multiple processors:

4 cpus initialize: Each processor selects a random number

IPI: inter processor interrupt, the kernel second stage bootloading

64 bit mode

Deliver Start up IPI

Hypervisor: Responsible for handling IPI

Goto sleep and wake uo when instruction comes

HLT: HALT IS USED

Until you receive an unmasked interrupt,

CLI

Stop all interrupts

Cli after HLT the cpu stops

Internal events :IPI

Interrupt f8 : exceptions Software exceptions interrupt, page fall, inavalid instruction, (exceptions) at the top

Os is responsible for acknowledging a n interrupt, disk controller ,

Software interrupts are also an other kind of interrupts

Irq107

Hardware interrupts knernel,

Software interrupt : system calls

MMU