



Introduction



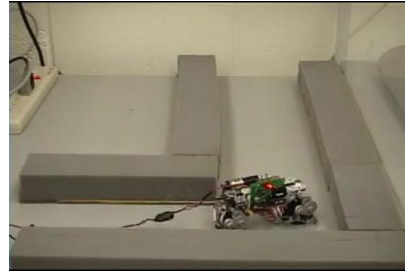
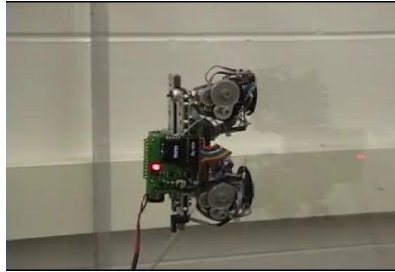
1

Warming Up

- Instructor Introduction
 - IBM 3090 vector processors supercomputer (precursor to modern GPU)
 - Security “plugin”: allow at most 4 users at any time to use the vector processors
 - **Lesson learned:**
 - Reentrant code at assembly level
 - Interrupt mechanism
 - [Miniature Climbing Robot](#) 🤖
 - **Lesson learned:** Build a mini OS: interrupt handler, CPU scheduler, memory manager, I/O
- Brief student Introduction
 - Name
 - Something unique about you (*to help me learn who you are*)

2

Miniature Climbing Robots



3

Announcements

- Access to EOS (Lab Thursday and subsequent weeks)
 - a. Use SSH Client + VPN required if you access the lab from off-campus
 - b. Use Web-based Portal (<https://computing.gvsu.edu/>)
- Course Web Site at <https://dulimarta-teaching.netlify.app>
 - a. Bb will be used mainly for announcements, collecting assignments, and grading

4

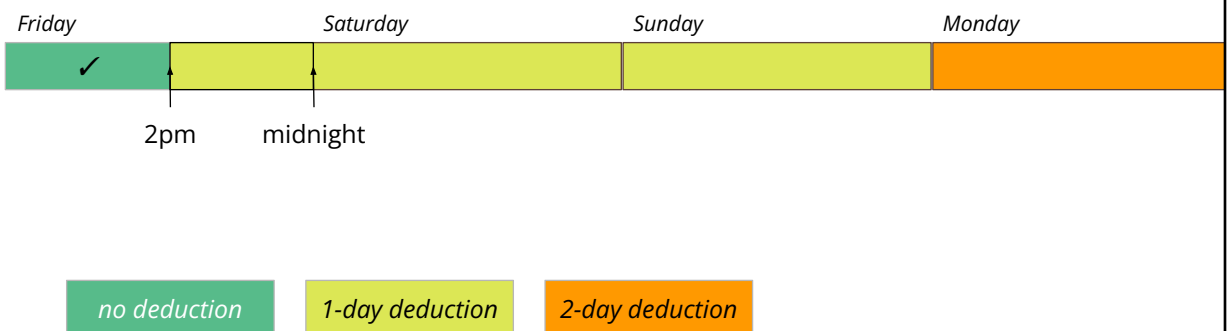
Grading/Late Policy

- All assignments are **due at the beginning** of class/lab time
- Each student has a **5-day late quota** throughout the semester
 - a. University holidays count as 0 day
 - b. Sat & Sun are logically the same day
- When your quota goes to zero, late assignments must be turned in no later than 3 days after due date. Apply 15% daily penalty
- No extra assignments will be provided per individual student requests. Use the extra credit opportunities provided in most assignments

5

Late Quota Deduction

Example: Assignment Friday due at 2pm



6

slido



What is your major

① Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

What is an OS?
A computer without an OS?

Demo: VirtualBox & Bare HW

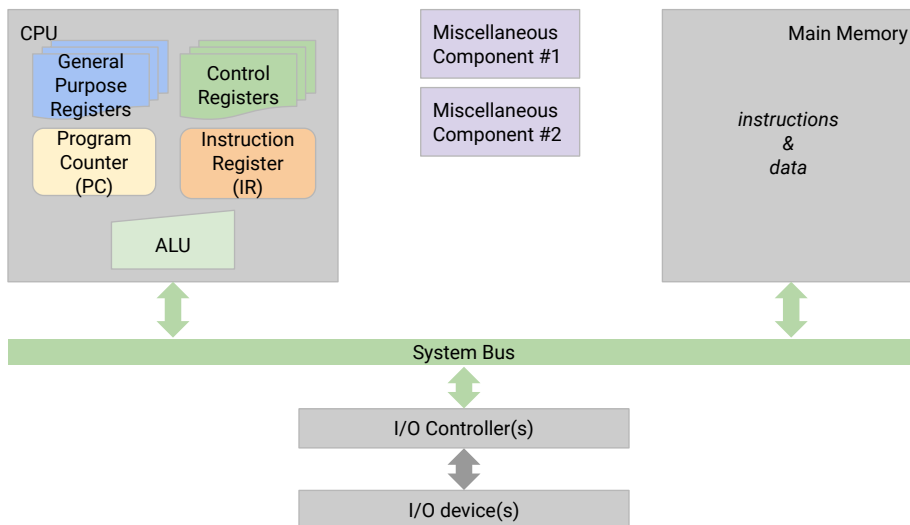
Basic Computer Hardware Elements

Basic Computer Hardware Elements

- Processor(s)
- Main Memory
- I/O Controllers & Devices
- System Bus

11

Basic Components of Computer Hardware



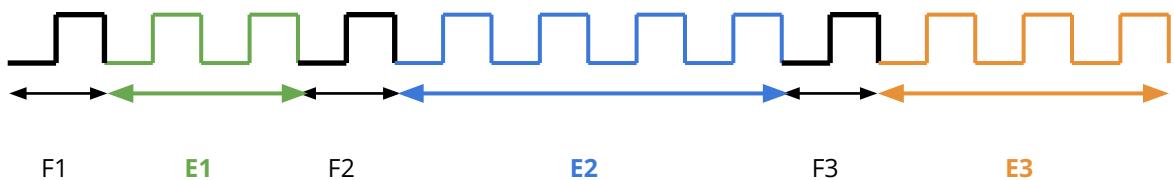
12

Important Fact/Concept

OS is NOT involved in fetching and executing every assembly instructions on your CPU(s)

18

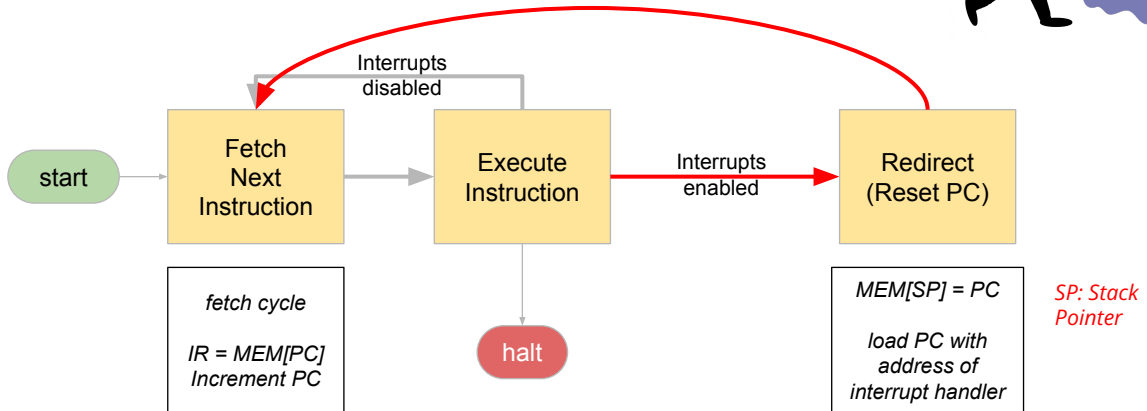
Microscopic View of Machine Cycles



How do you interpret the diagram?

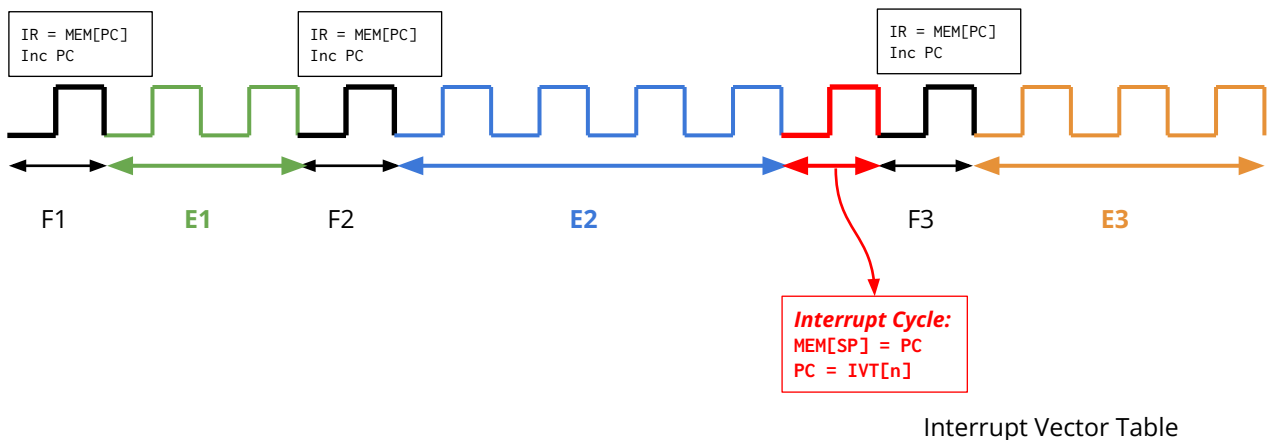
19

Instruction Cycles with Interrupts



CPU instruction cycles: FEF EFEFE **I** FEF EFEFEFEFEFEFEFEFE....

Microscopic View of Machine Cycles **with Interrupt**



Interrupt Vector Table (IVT)

The table begins at a **known location** in RAM



00	Addr of code for handling INT 00
01	Addr of code for handling INT 01
02	Addr of code for handling INT 02
255	Addr of code for handling INT FF

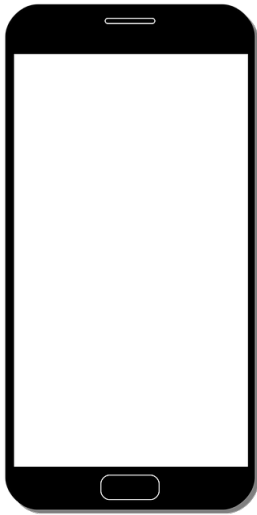
24

Interrupts

- *Misconception:* CPU sends hardware interrupt signal to your program
- Fact:
 - a hardware interrupt: signal **asynchronously** received by CPU sent by devices/other units
 - a software generated interrupt: **synchronously** triggered by a special assembly instruction, or errors detected during execution of the current assembly instruction
 - In response: CPU "immediately" enters its interrupt cycle to service the interrupt request
 - Side effect: the current program is interrupted
- Source of interrupts
 - (Async) Triggered by other components in your computer
 - Real-time clock, keyboard, gyroscope (smartphone), ...
 - (Sync) Detected by CPU: Arithmetic errors, unknown instructions, protection error,
 - (Sync) Requested by **software** (INT or TRAP instruction)

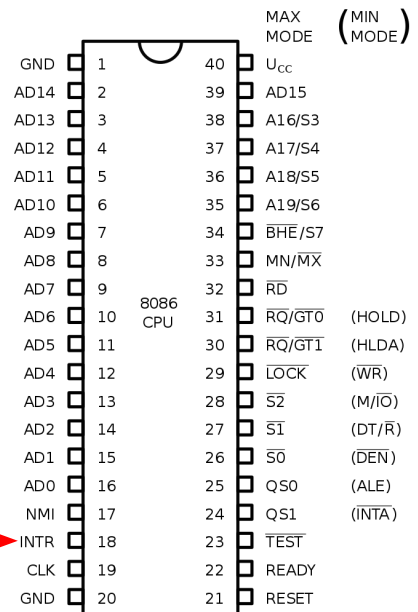
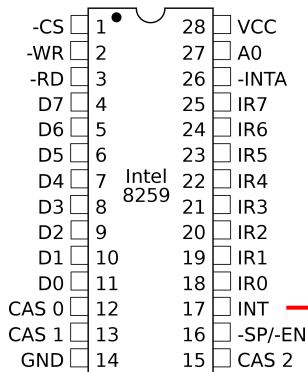
25

MANY different sources of hardware interrupts

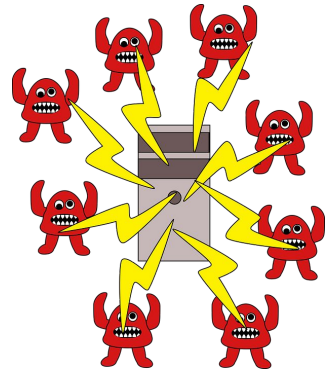


Gyroscope on your phone can send hardware interrupts

PIC: Programmable Intr. Ctrl.



How does ~~the OS~~ HW protect itself from malicious programs?



31

Assembly Code Demo: Syscall and HLT (on VMWare Fusion or Docker Container)

File: OneDrive-GVSU/CS452-xxx/Container/Debian

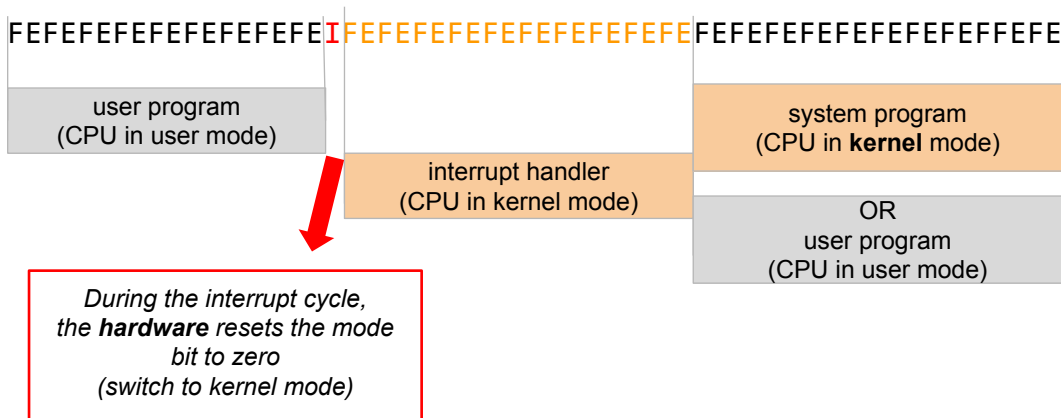
32

User Mode vs. Kernel Mode

- Certain CPU instructions are privileged (*should execute only when the CPU is in kernel mode*)
 - Examples: **HLT** (*to stop the CPU*), **CLI** (*the disable interrupt*), ...
- CPU keeps a mode bit in one of its *control registers*
 - mode = 0 (kernel mode): CPU can execute **all the available instruction sets**
 - mode = 1 (user mode): CPU can execute **only non-privileged subset** of CPU instructions
- Obviously, setting the mode bit to 0 cannot be done while the CPU is in user mode. Solution?

33

Dual Mode Execution: Kernel Mode vs User Mode



34

Misconception: *User/Kernel mode of the CPU is determined by your login as either a normal user or root/admin user*

Fact: *The CPU mode is unrelated to user access level (admin/root)*

- CPU mode is a feature provided by hardware
- Admin/root user state is managed in software