CS 342302 Operating Systems

Fall Semester 2021

Prof. Pai H. Chou

Weekly Review 1

The questions here serve the purpose of reviewing concepts from the lecture, and expect the concepts to be tested on the midterm and final. However, they are by no means exhaustive. Anything covered in the lecture and projects can be tested.

1. Definitions and Short Answers - week 1 (9/13 lectures)

1. What is **batch processing**? What are its advantages? Disadvantages?

Advantages :

Repeated jobs are done fast without user interaction.

Offline makes less stress on processor

Sharing system for multiple users

You can assign specific time for the batch jobs

Drawbacks :

one job at a time

no interaction between users and jobs

CPU is often idle: I/O speed much slower than CPU speed

1. What is **multiprogramming**? What disadvantage of batch processing does it address?

p27

1. Compare **multiprogramming** and **multitasking** in terms of number of users, number of jobs running, and need for support features. (textbook p. 23)

1, more, memory management, CPU scheduling, IO system

more, more, virtual memory, storage, file system, synchronization, deadlock

1. What is an **instruction set architecture** (ISA)? How is it different from a CPU or a processor?
2. What are reasons for the trend from single processor to multiprocessor architectures?

throughput, economical, reliability, power efficiency

1. What makes tightly coupled multiprocessors difficult to **scale** to a large number of processors?

consumes too much energy

1. What are examples of **real-time** systems? How do they differ from **high-performance** systems?

p46

1. What are examples of **hard real-time** vs **soft real-time** systems?

p47

2. EdSim51 and 8051 - week 1 (9/15 lecture)

1. What is an assembler?

translate assembly to machine code

1. Given the sample assembly program:  
   ORG 0000H  
   MOV 90H, #24H  
   END
   1. What is a **directive** in this assembly program, and how is it different from an instruction?

information only for assembly code, wont be translated into machine code

* 1. What is an **instruction**? What does the assembler do to an instruction?
  2. What is an **opcode** in the above example? An **operand**?

MOV

90H, #24H

1. What does 90H refer to? What does #24H refer to? Why is there a # in front of 24H but not in front of 90H?

address, literal value, mark 24H as literal value

1. What is a **NOP**? and how do you pronounce it?

no operation

1. Why is 8051 called a **Harvard architecture**? How is it different from a **von Neumann** architecture?

it has external & internal memory

1. What is a **PC** in a processor? What are the ways PC value can change?

program counter, jump instruction

1. How many bytes is 8051’s **code memory**? How many bits are needed to represent the code address?

64KB, 16bits

1. What is **DPTR** in 8051? How is it related to **DPL** and **DPH**?

data pointer, DPTR=concat(DPL+DPH)

1. How big is the IDATA memory in 8051?

256Bytes

1. What is a special-function register (SFR)?

the upper area of idata, can't be used for data or program storage

it is for special function

1. What is the meaning of **simplex**, **half-duplex**, and **full-duplex** communication? Which one is the UART (serial port)?

duplex: can do 2 way communication

simplex: 1 direction only

half-duplex: 1 direction at a time, support both direction

full-duplex: 2 way communication at the same time

UART => full-duplex

1. if #24H refers to hex 24 integer value (“immediate”) in Intel assembly, why #FFH does not refer to hex FF integer value? Name two ways to express 0xFF in Intel assembly syntax.

The char after # must be 0-9

append 0 at front, -1

1. What is the meaning of MOV A, 17 in 8051 assembly, and how is it different from MOV A, #17 ? What about MOV A, 17H ? MOV A, #17H ?

MOV A,17: A = \*((char\*)17)

MOV A,#17: A = 17

MOV A,17H: A = \*((char\*)0x17)

MOV A,#17H: A = 0x17

1. What is a **general-purpose input/output** (GPIO) port? What are they called on 8051?

a grouping of 8 pins that can be configured for output or input

P0, P1, P2, P3

1. Why do you have to write a **0 bit** to **turn on** an LED segment? Why write a **1 bit** to **turn off**?

0 means gnd => have voltage diff => led on

1. What is an “immediate” operand?

constant value

1. What is a "direct" operand?

operand that specifies the direct address

1. How do you pronounce “UART”? What is another more descriptive term for UART?

Universal Asynchronous Receiver/Transmitter

serial port(TxD, RxD)