# CS1356 Introduction to Information Engineering

## Homework 6

Read textbook 3.1 and answer the following questions

1. What is a **job**? How to execute a program when there is no operating system?

2. What is a **batching system**? Why does it use a queue to scheduling the job execution? What are the properties of a queue? What would happen if the job execution is according to priorities?

3. What is a **job control language** (JCL)? Give an example that you had seen on your desktop.

4. What is the drawback of a batch system?

5. What is an **interactive process**? Give two examples that require iterative execution.

6. What is a **real-time process**? Give two examples that have real-time requirement.

7. What is the difference between a real-time process and an interactive process?

8. How to make a computer to execute two real-time processes simultaneously?

9. What are **time-sharing**, **multiprogramming**, and **multitasking**?

10. What is the difference between time-sharing and multitasking?

11. What is the system architecture of a multiuser, time-sharing system?

12. What is the role of modern **operating system** (OS)?

13. What are the **load-balancing problem** and the **scaling problem**?

14. What are the impacts of network systems to the design of an OS?

15. What are the challenges of OSs on small devices like cell phones or PDAs?

Read textbook 3.3 and answer the following questions.

1. What is a **program** and what is a **process**? What is their relation?

2. What is a **process state**? What kinds of information should be recorded in a process state?

3. What are computer's resources? Give three examples.

4. What is the job of a **scheduler**? And what is the job of a **dispatcher**?

5. What is the **process table**? Which information is in the table? When to add an entry to the table?

6. What does it mean when a process is said being **ready** or being **waiting**?

7. What is a **time slice**? Which occasion needs to define a time slice?

8. What is an **interrupt** in a computer system? What can trigger an interrupt? And how does an OS respond to an interrupt.

9. What is the effect of the interrupt signal trigger by the timer? What would the dispatcher do when receiving an interrupt?

10. What is the most important ability of a process in a multiprogramming system?

11. Why does an OS need to recreate the process's state when the process is restarted? And what should it do to prepare this recreation?

12. Summarize the process of running two tasks simultaneously.

13. What is the overhead of multiprogramming? With such performance overhead, why do most existing OSs use multiprogramming to manage processes?

14. If each time slice in a multiprogramming system is 50 milliseconds and each context switch requires at most a microsecond, how many processes can the machine service in a single second? What fraction of the machine's time is spent actually on performing processes? What would this fraction be if each process executed an I/O request after only a microsecond of its time slice?