This document lists some points that may be covered in the exam, but you should not consider this as a comprehensive list, but a guideline on what may appear in the exam questions.

- OS revolutions
  - What features each generation supports, what the driving forces are, what enabling techniques are required to support its features, etc.
  - May create a mock OS and ask you which techniques need to be supported (e.g., context switch, interrupts (which types of interrupts), etc.)
  - May give a scenario or some requirements and ask you which OS would be most suitable (or **rank** the choices)
- The instruction cycle and each phase in the instruction cycle
  - Micro-instructions that shows what actions are taken when handling each instruction
- Different types of interrupts
  - How interrupts are handled, what are the impacts of interrupt handlers
  - When a certain type of interrupt would be raised, what actions are to be taken when handling each type of interrupts, etc.
- Context switch
  - What program state information (registers) needs to be saved
  - What overhead it causes, etc.
- Process states
  - What states a process may go into, what each state meant
  - What triggers each state transition, what corresponding actions (major ones) are to be taken
  - Questions like Q1 in hw1
- May mix multiple concepts above to test your real understanding
  - May give you a program of regular statements as well as system calls, and/or a list of events happened during its execution and ask you to decide which statements/events may trigger some interrupts, context switches, process state transitions, etc. and what will happen after the interrupts, context switches, process state transitions are triggered
  - New mock instances may be added to ensure that you understand the concepts

- Difference between processes and threads, their creation mechanisms, execution mechanisms, the sources of overheads, overhead comparisons, etc.
  - E.g., compare some specifics after a process and/or a thread has been created
  - E.g., compare the results after partial or complete executions
  - E.g., compare the overheads for their creation, execution, and termination
  - E.g., give a piece of code and ask you what have been created and some specifics in what have been created, what output may be generated, etc.
- Process control blocks, what it is, what entries are needed, etc.
- Thread control blocks, what entries are needed, etc.
  - May give a list of items and ask you to decide which one should be included in PCB or TCB
  - May give a piece of code and ask you what changes may have to be made to the PCB/TCB

- Understand the inter-process communication mechanisms, such as pipes, signals, and sockets (socket will be for the next exam)
  - You do not need to remember syntactic information for these functions, but you need to understand their creation mechanisms (what happens in the system), and their execution behaviors
  - May give a program (could be with some problems) and ask you what behaviors may occur
  - May ask specific questions about their behaviors

- Understand the problems with concurrent programming in shared memory model
- Given a concurrent program, what potential behaviors the program may have
  - Whether mutual exclusion, synchronization requirements, etc. will be satisfied
  - Whether there will be starvation, deadlock, after you after you, etc. problems

    - What are the possible values of some variables (or multiple variables) or what may be the outputs
    - Note: when we do not mention the level of atomicity, it should be the actual level of atomicity in all systems, i.e., the instruction level, not the statement level

- Understand lock, semaphore, and monitor (monitor is excluded)
    - How to implement them
        - You may be given an improper implementation and asked to determine what may happen if used for some purposes
    - How to use them
        - Use them to achieve mutex and sync goals for a given problem and given sync/mutex requirements (pseudo code)
        - Put some statements in a program to achieve the desired sync/mutex
    - Pros and cons
        - Pros and cons in general
        - Pros and cons for using them for mutex and sync, especially under specific scenarios or for a specific program
        - Compare them for specific scenarios or in a specific program
    - Additional considerations
        - For lock, there are software and hardware solutions, consider the differences, the pros and cons
        - For semaphores, it can be integer or binary semaphores, what would be the differences in using them?
        - For monitor, there are some special issues to be considered in implementing the supporting code that you need to study
- Understand the mechanism of disable interrupt, its capability, its limitations
    - Need to understand how to use it
    - Need to know the pros and cons of using it

About the exam and auto-grading:

- Check survey to see some instructions for the exam
- Coverage and readings
    - Based on the 9th Edition
    - Reading list at the end of each slide set
    - For the last slide set, we did not cover 5.5, 5.6 yet