(乙組)

Part I. Operating Systems (50%)

1. (10%) What advantages does segmentation have over paging? What problems arise in sharing code in segmentation schemes?

2. (10%) In the table below eight jobs to be scheduled on two identical processors, the times at which these jobs become available, and their required processing times are listed. Assume that jobs can be scheduled instantly.

Job	Available	Processing Time
A	0	6
B	0	2
C	0	3
D	2	5
E	3	4
F	5	1
G	7	3
H	9	6

(a) What is the earliest time at which processing of all jobs can be completed?
(b) Assume that the criterion for scheduling is to minimize the delay in starting the processing of each job and assume no preemption. What is the minimum average delay in starting time?

3. (5%) Answer true or false to the following question and explain your reason.

Mutual exclusion requiring busy wait is a waste of cpu resource. By using semaphores and the wait and signal operations we can eliminate busy wait.

4. (10%) Define the following terms: (1) rendezvous (2) daemon process

5. (5%) The swap hardware instruction swaping the contents of two words and being available in some systems can be used to solve the critical-section problem. Write a software swap procedure which in a uniprocessor environment has the same effect as the hardware swap instruction.

6. (10%) The protection scheme in the Unix system is basically designed around the file system. Discuss the Unix file protection scheme, especially for domain switching.

(乙組)

Part II. Compilers (50%)

1. (10%) One of the important functions of a compiler is the detection of errors in the source program. For each of the following phases of a compiler, state the type of errors detected. Give an example for each type of errors.
(a) lexical analysis (b) syntax analysis (c) intermediate code generation (d) code optimization (e) code generation

2. (10%) In certain programming languages (e.g. Fortran) in order to identify a particular token (e.g. DO) the lexical analyzer requires to "lookahead" some extra characters beyond the token itself. State in such a language how to use "lookahead operator" to specify the particular token and how to implement the "lookahead operator" and the lexical analyzer such that the correct scanning of the characters stream can be promised.

3. (5%) The following syntax-directed translation scheme is used with a shift-reduce (bottom-up) parser that performs the action in braces immediately after any reduction by the corresponding production.

```
A -> aB    {print "0"}
A -> c     {print "1"}
B -> Ab    {print "2"}
```

What is the string printed when the parser input is aacbb?

4. (10%) Answer the following questions.

(a) (true/false) The LR(1) table for a grammar has a shift/reduce error if and only if its LALR table has one.

(b) (true/false) If a grammar produce a shift/reduce conflict when run through YACC then it is ambiguous.

(c) (true/false) Operator-precedence parsing is a kind of bottom-up parsing.

(d) (true/false) The following grammar is not a LL(1) grammar

```
S -> SB | SC | ϵ
B -> aBb | ab
C -> bCc | bc
```

(e) The activation record consists of a collection of fields saving information. Two of which are for saving control link and access link. Briefly explain the purposes of these two fields.

5. (10%) Explain the mechanism provided by YACC for resolving all parsing action conflicts.

6. (5%) Define "peephole optimization" and give an example to support your explaination.