KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

B.Sc. Engineering 3rd Year 2nd Term Examination, 2015

Department of Computer Science and Engineering

CSE 3201

Operating Systems

TIME: 3 hours FULL MARKS: 210

N.B. i) Answer **ANY THREE** questions from each section in separate scripts.
ii) Figures in the right margin indicate full marks.

SECTION A

(Answer **ANY THREE** questions from this section in Script A)

1. a) Define distributed system. Discuss the merits of it. (08)
   b) What are the types of distributed operating systems? Discuss them. What are the reasons to use 'process migration'- Explain them. (09)
   c) Discuss the most common routing schemes in distributed communication with possible merits and demerits. (09)
   d) What are the techniques to avoid repeated collisions over a communication network? Explain them. (09)

2. a) Discuss the characteristics of deadlock. Depict a deadlock situation with an appropriate Resource-Allocation Graph. (09)
   b) What are the methods to handle deadlocks? Discuss the pre-conditions and data structures for Banker's Algorithm. Also explain Banker's Algorithm with an example. (17)
   c) How a deadlock in case of single instance of each resource type is handled using wait for graph? Discuss the issues of resource preemption to recovery from deadlock. (09)

3. a) What is meant by 'man-in-the-middle' attack? Discuss how a boot-sector computer virus affects the operating system. (10)
   b) RSA cryptosystem can ensure secure communication over insecure medium-how? Explain with an example. (09)
   c) How digital signature can ensure user authentication? Explain. (07)
   d) Define access matrix. How it can be implemented? (09)

4. a) With necessary diagrams explain how instructions and data can be combined to memory address. Why are page sizes always power of 2? (10)
   b) Discuss the problems in case of contiguous memory allocation. Dynamic storage allocation problem can be solved in many ways- explain any one of them. (10)
   c) Define paging. Give an example of it. (07)
   d) How TLB facilitates to implement paging hardware? Explain with a diagram. (08)

SECTION B

(Answer **ANY THREE** questions from this section in Script B)

5. a) Why should we study operating systems? What do you mean by system program and loadable kernel module? (10)
   b) What is process state? Write down the process states briefly with proper diagram. (07)
   c) How does CPU switch from process to process? Explain with proper diagram. (10)
   d) Write a pseudo code for creating child process form parent. How can you make this child orphan? Explain with pseudo code. (08)

6. a) Define the following terms: (06)
      i) Data parallelism, ii) Task parallelism and iii) Concurrency.
   b) "Serial portion of an application has disproportionate effect on performance gained by adding additional cores"- explain this statement using Amdahl's law. (08)
   c) What do you mean by thread library and thread pool? Describe two level model of multithreading. (10)
   d) What is priority scheduling? What is the main problem and corresponding solution of priority scheduling? (06)
   e) What is exponential averaging? (05)

7. a) Suppose we have following processes and their corresponding arrival and burst time: (15)

Process	Arrival time	Burst time
P_1	0	2
P_2	1	7
P_3	3	9
P_4	4	1
P_5	5	8
P_6	7	2
P_7	9	5

Draw the Gantt chart for shortest remaining time first algorithm. Find out average waiting and turn-around time.

b) Suppose there are two processes P_1 and P_2. P_1 has period $P_1 = 50$ and CPU burst $t_1 = 25$ and P_2 has period $P_2 = 80$ and $t_2 = 35$. Is it solvable by rate monotonic scheduling? Why or why not? Solve this problem with earliest deadline first (EDF) scheduling and draw necessary Gantt chart. (13)

c) What is little's formula? Explain briefly with an example. (07)

8. a) What is busy waiting? Explain it using strict alteration method. (08)

b) Solve the producer-consumer problem with semaphore. Write down necessary semaphore structure, wait and signal method, necessary conditions and code for producer and consumer process. (15)

c) What is contiguous allocation of file? Write down the advantages and disadvantages. (07)

d) Discuss about distributed file system. (05)

KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY
B.Sc. Engineering 3rd Year 2nd Term Examination, 2015
Department of Computer Science and Engineering
CSE 3211
Compiler Design

TIME: 3 hours FULL MARKS: 210

N.B. i) Answer **ANY THREE** questions from each section in separate scripts.
ii) Figures in the right margin indicate full marks.

SECTION A

(Answer **ANY THREE** questions from this section in Script A)

1. a) Differentiate between token, pattern and lexeme with example. (08)

   b) What are the phases of a compiler? Explain the syntax and semantic analysis of the statement z = 4*a + b*i. (12)

   c) Write a program in flex to recognize identifiers (start with letters followed by letters and digits), assignment operator (:=), arithmetic operators (+, -, *, /), relational operators(<, >, =<, >=), key words (if, else, while, for) and comments (// and /*...*/). (15)

2. a) Consider the following code segment (15)

```
void r(){ int i; ...}
 int p(int m, int n){...}
void q(int m, int n) { int i;
   if(m>n){ i=p(m, n); q(m, i-1); q(i+1, n);} }
main(){ r(); a[0]=-999; a[10]=999; q(1,9);}
```

   i) Define activation tree, activation record, and control stack
   ii) Draw the activation tree for the above code segment
   iii) Show the control stack at q(2,3).

   b) What is dangling else problem? Explain with example. (07)

   c) What is the idea of left factoring of a grammar? Apply left factoring in the following grammar A->ad/a/ab/abc/b (08)

   d) What are the problems of top down parsing? (05)

3. a) What can be the contents of the stack for LL(1) parser? What are the actions taken by the parser if the top of stack is a non terminal X? (09)

   b) Consider the general configuration of a LR parser $(S_0X_1S_1\ldots X_mS_m,\ a_i\ a_{i+1}\ldots a_n\ \$)$. Explain the actions by the parser based on $< S_m, a_i >$. (09)

   c) Find the canonical collection of sets of LR(0) items for the following grammar. (12)

   $E' \to E \quad E \to E + T / T \quad T \to T*F / F \quad F \to (E) / id$

   d) What will you do if the resulting table for LL parser contains multiply defined entries? (05)

4. a) Define intermediate code. How can you represent intermediate code? (06)

   b) Consider the following code segment (18)

   i = 2*n + k; while (i) do i = i-k;

   i) Generate the three address code
   ii) Implement the three address code using quadruples and triples
   iii) Write a semantic rule for while statement

   c) How can you translate the switch-case statement into three address code? Write the translation scheme for the following switch-case statement (11)

```
switch(a+b-c) { case 1: z = x+y; break;
                case 2: z = x-y; break; }
```

SECTION B

(Answer **ANY THREE** questions from this section in Script B)

5. a) Define look ahead sysmbol. What are the properties of a parse tree? (07)

   b) Suppose a desk calculator reads an input line containing an arithmetic expression involving digits, parentheses, the operators '+' and '*' followed by a new line character '\n' and prints ***the value of the expression.*** (10)

   i) Write down the syntax directed definition for the calculator
   ii) Draw the annotated parse tree for the input 3*5 + 4*n

   c) Suppose a declaration consist of the key word **int** or **real** followed by a list of identifiers (10)

   i) construct the grammar for the above declaration
   ii) Draw the dependency graph for the input **int a, b, c.**

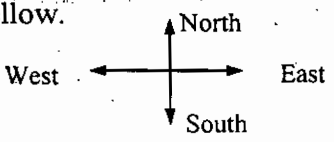
d) "A syntax tree is acondensed form of a parse tree"- justify the statement with proper example. (08)

6. a) What does the static check mean? Describe some static checks with examples. (07)

b) What is type expression? How is the type constructor applied to type expressions to get type (08) expression for arrays, records and functions?

c) Construct the Directed Acyclic Graph for the statement a+a*(b-c)+(b-c)*d+a*(b-c) (08)

d) Suppose a robot can be instructed to move one step east, north, west or south from its initial (12) position as shown bellow.



i) construct a grammar for movement of the robot

ii) To verify your grammar show an input string that can be obtained from the grammar.

iii) Draw annotated parse tree for an input. All the sample input must begin with an initial position start at (-5, 3) which is followed by at least four different directions.

7. a) Briefly describe following peephole optimizations with proper example: i) Redundant (08) instruction elimination ii) Algebraic simplifications.

b) A language is defined by the following grammar P→D;E D→D;D/id:T (10) T → char/integer/array[num] of T E→ literal / num / id / E mod E / E[E]
Design a type checker for the above language

c) "Statement by statement code generation often produces poor code" – justify the statement (07) with proper example.

d) Consider the following code segment (10)

```
/* code for C */          /* Code for P */
  action 1                  action 3
  call P                    return
  action 2
  halt
```

The code for the procedures starts at addresses 200 and 400 respectively and each action instruction takes 40 bytes. The activation records for the procedures are statically allocated starting at location 600 and 664 respectively. Show the static allocation for the code segment.

8. a) "Applying one optimization may raise opportunities for other optimizations"- justify the (08) statement with proper example.

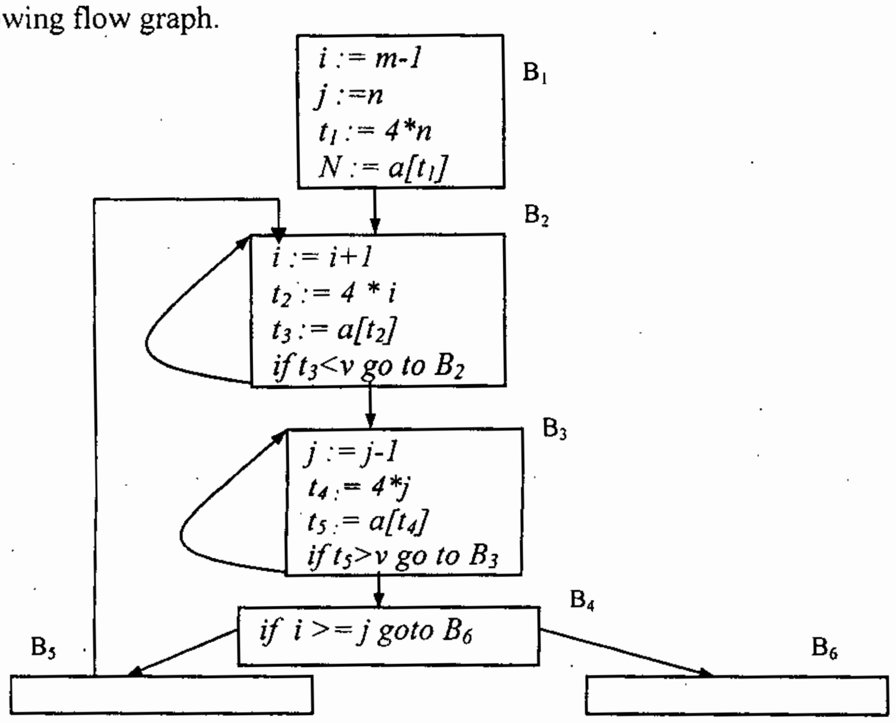
b) Eliminate common sub expression from the following code segment (05)

*up = v[(i-1)*n + j]; down = v[(i+1)*n + j];*
*left = v[i*n + j-1]; right = v[i*n + j + 1]; sum = up + down + left + right;*

c) Calculate the cost of the following instructions (12)

```
i)   MOV B, R0    ii) MOV B, A    iii) MOV *R1, *R0
     ADD C, R0        ADD C, A         ADD *R2, *R0
     MOV R0, A
```

d) Apply the techniques induction variables elimination and reduction in strength on the (10) following flow graph.



```
B1:
i := m-1
j :=n
t1 := 4*n
N := a[t1]

B2:
i := i+1
t2 := 4 * i
t3 := a[t2]
if t3<v go to B2

B3:
j :=j-1
t4 := 4*j
t5 := a[t4]
if t5>v go to B3

B4:
if i >=j goto B6
```