20115102

Mid-Semester Examination (CBCS), B.Tech. (V Semester, Session 2022-23)

Subject: Operating Systems **Branch: Computer Science & Engineering**

Time: 2 hours **Max Marks: 30**

Note: Attempt any **Five** Question. Q.1 is Compulsory. All Questions carry equal marks.

Q. 1 Consider the following set of processes with the length of CPU burst time given in milliseconds: What will be the average Turnaround and average waiting time for these processes using SJF, SRTF and Round Robin(q=2) Scheduling algorithms?

Process	Arrival time	Burst Time
P1	0.0000	4
P2	2.0001	7
P3	3.0001	2
P4	3.0001	2

Q.2 (a) Explain seven state models of process and show transition between process states, indicating whether or not the transition is possible. If it is possible, give an example of one thing that would cause it.

(b) What are threads? What resources are used when a thread is created? How do they differ from those used when a process is created?

Q.3 Define the following terms: (i) Spooling (ii) Scheduler (iii) Kernel (iv) Process control block (v) Dispatch latency (vi) Convoy effect

Q.4 Explain different directory structures of operating systems with their respective strengths and limitations.

Q.5 Suppose that the disk request queue contains requests for sectors on the following cylinders: 400, 20, 19, 74, 899. Suppose that the current position of the disk arm is over cylinder 200. In which order will the requests be handled using following disk head scheduling policy (i) SSTF (ii) SCAN (iii) C-SCAN (iv) LOOK (v) C-LOOK (vi) FCFS (assuming the initial direction is up) 0-1000

Q.6 (a) What is file? Explain different attributes, types and operations of a file.

(b) Explain various disk space allocations methods with their advantages and disadvantages.

20115052 X

NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR

B.Tech. (5th Semester) Mid-Semester Examination, Autumn 2022

Subject: Compiler Design (Code: CS105101CS), Branch: CSE

Time: 2 Hours **Max Marks: 30**

Note: Answer any five including Q1. (Q1 is Compulsory)

Q1. Answer the following questions **[1x10]**

a) Eliminate left recursion from the given grammar.

$S \rightarrow Aa \mid b$

$A \rightarrow Ac \mid Sd \mid \lambda$

b) Design context free grammar for the language which is the set of all strings of 0's and 1's with an unequal number of 0's and 1's.

c) Give a RE and a CFG for: $L = \{x \in \{0, 1\}^* \mid x \text{ starts and ends with different symbols}\}$.

d) What is the precedence and associativity of operators &, @, #, ↑in the following grammar?

E→ T↑E|T

T→T&F|F

F→P@F|P

P→#R|R

R→id

e) Eliminate left recursion from the grammar:

A→b|Bd; B→Bc|Ac

f) Check whether the grammar S→AB, A→a|λ, B→b|λ is LL(1) or not? Justify.

g) Find the number of tokens in the flowing C statement

```
int a,b;
/* finding max of a and b*/
int max(a, b) {
return a> b? a: b; }
```

h) Consider line number 3 of the following C-program

```
int main(){
int i, n;
fro (i=0; i < n; i++); }
```

Identify the compiler's response about this line while creating object module

i. No compilation error

ii. Only a lexical error

iii. Only a syntactic error

iv. Both Lexical and syntactic errors

i) What is the follow (Q) in the following grammar?

P→xQRS

Q→yz|z

R→w|λ

S→y|λ

j) Match the following

LIST-I		LIST-II
v.	Lexical Analysis	e. Type checking
vi.	Top Down Parsing	f. Regular Expression
vii.	Semantic Analysis	g. Left most derivation
viii.	Runtime Environment	h. Activation Records

Q2. a. Give a CFG for: $L = \{0^i1^j0^k \mid j > i + k\}$ So, 001111100 is in the string. [2.5]

(Hint, the concatenation of two (or more) context-free languages is context-free.)

b. Consider a CFG with {S, A, B} as non-terminals, {a, b} as terminals, {S} is the start symbol and the following set of rules {S→AB, S→bA, A→a, A→aS, A→bAA, B→b, B→bS, B→aBB} [2.5]

I. Which of the following string is generated by the above grammar?

i. aaaabb ii. aabbab iii. aabbbb iv. abbbba

II. For the correct answer above problem, how many derivation trees are there?

i. 1 ii. 2 iii. 3 iv. 4 v. None

Q3. What is the advantage of dividing the design of a compiler into front-end design and back-end design? Show the output of all phases of the compiler for the following lines of code x=a+b*c. [5]

Q4. a. Left Factor the grammar

S→ iEtS |a| iEtSeS

E→b

Find the first() and follow () set of the left factored grammar. Construct the predictive parsing table for the resulting grammar? Check the grammar is LL(1) or not? If not find the position of multiple entries in the table? [2.5]

b. Construct LL(1) parsing table for the following grammar [2.5]

S→aBDh

B→cC

C→bC|λ

D→EF

E→g|λ

F→f|λ

Check whether the Given Grammar is LL(1) or Not. [5]

Q5. For the following grammar

E→E+T | T

T→T*F |F

F→(E) |id

i. Eliminate left recursion from the grammar rules.

ii. Find first () and follow () set for the resulting grammar.

iii. Construct predictive parsing table for the modified grammar.

iv. Show the actions of the parser for the input string "id*id+id".

v. Draw the parse tree. [2x2.5]

Q6. Write short notes any two

i. Symbol Table

ii. Components of a language processing system

iii. Recursive Descent Parsing

2011503

NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR

B.Tech (5th Semester) Mid Semester Examination, Autumn 2022

Subject: Data Science (Code: CS105302CS)

Branch: CSE/ELE/ECE/ITY/MEC/MME/CHE/CIV/BMD/BOT/MIN

Time: 2 Hours **CBCS SCHEME** **Max Marks: 30**

Note:

(1) All questions from Section-I are compulsory.

(2) Solve any four questions from Section-II.

(3) Assume suitable data if necessary.

Section-I (All questions are compulsory)

Q1. Short Answer Questions

a) Is the given function, a valid cumulative density function (CDF) for discrete random variable x? F(x) = (x - 4)/6; 4 <= x <= 10. [2]

b) Explain the term "Big Data". What are the different V's of Big Data? [2]

c) Define terms variance and standard deviation. There are 5 sample observations given for a population: 4,2,5,8,6 calculate variance and standard deviation. (Show all calculation steps) [2]

d) Find the maximum, minimum, median, first quartile, and the third quartile for the given data set. Also, draw an approximate box plot showing all these values in the plot. [2]

Person	1	2	3	4	5	6	7
Age	23	42	12	10	15	14	9

e) *For the given data find the* normalized value of 19 using the following normalization *techniques.* [2]

I. *min-max* normalization (Range 0 to 1)

II. *z-score* normalization

Data: -

3	5	5	8	9	12	12	13	15	16	17	19	22	24	25	134

Section – II Long Answer Questions (Attempt any Four Questions)

Q2. Explain different moments of a continuous random variable. [5]

Q3. Explain pattern recognition architecture in detail. [5]

Q4. Explain CDF and its properties in detail. If a person tosses the coin three times, let X be the number of the observed tail. Find the CDF of X. [5]

Q5. What is data pre-processing? What are the different steps and techniques involved in data pre-processing? [5]

Q6. Obtain the regression equation of Y on X and estimate Y when X=55 from the following. [5]

X	40	50	38	60	65	50	35
Y	38	60	55	70	60	48	30

Q7. Calculate the covariance and co-relation between the stocks INFY and TCS. [5]

Year	INFY (Stock Price)	TCS (Stock Price)
2013	1692	68
2014	1978	102
2015	1884	110
2016	2151	112
2017	2519	154

Q8. A pharmaceutical lab states that a drug causes negative side effects in 3 of every 100 patients. [5]
To confirm this affirmation, another laboratory chooses 5 people at random who have consumed the drug. What is the probability of the following events?

(a) None of the five patients experience side effects (up to two places of decimal).

(b) At least two had side effects (up to four places of decimal).

Q9. Suppose the number of times a web site crashes in any time interval are a Poisson random variable. A particular site on average crashes 6 times per day. What is the probability that there will be no crashes in an interval of two days? [5]

NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR
B.Tech. (FIFTH SEMESTER)
Mid Semester Examination, Autumn 2022
Department of Computer Science and Engineering
Program Elective – 1 (Advanced Computer Architecture (CS105202CS))

Date: 23-09-2022 (Time 10:00 AM to 12:00 PM) **Max Marks: 30**

INSTRUCTIONS:

- **Attempt any five questions out of the six questions. All parts of a question should be answered together.**
- **Marks assigned for each question are mentioned in the right-hand side.**
- **Notations have their usual meanings. Assume default values if required.**
- **You are allowed to use calculators.**

Q1. Please respond to parts (i) and (ii) in no more than 500 words.

(i) Indicate whether each of the following statements is **true or false** and justify your answer with reasoning and supportive or counter examples: **[2]**

(a) The CPU computations and I/O operations cannot be overlapped in a multiprogrammed computer.

(b) Synchronization of all PEs in an SIMD computer is done by hardware rather than by software as it is often done in most MIMD computers.

(c) As far as programmability is concerned, shared-memory multiprocessors offer simpler interprocessor communication support than that offered by a message-passing multicomputer.

(d) In an MIMD computer, all processors must execute the same instruction at the same time synchronously.

(e) As far as scalability is concerned, multicomputers with distributed memory are more scalable than shared-memory multiprocessors.

(ii) Define the following terms related to parallelism, dependence relations, and system interconnect architectures: **[4]**

(a) Bernstein conditions (b) Non blocking networks (c)Network diameter
(d) Bisection bandwidth (e) Grain packing (f)Static connection networks vs Dynamic connection networks
(g) Compare control-flow, dataflow, and reduction computers in terms of the program flow mechanism used.

Q2. (i) Characterize the architectural operations of SIMD and MIMD computers. Distinguish between multiprocessors and multicomputers based on their structures, resource sharing, and interprocessor communications. Also, explain the differences among UMA, NUMA, COMA and NORMA computers. **[4]**

(ii) Explain how instruction set, compiler technology, CPU implementation and control, and cache and memory hierarchy affect the CPU performance and justify the effects in terms of program length, clock rate, and effective CPI. **[2]**

Q3. (i) A 40-MHz processor was used to execute a benchmark program with the following instruction mix and dock cycle counts:

Instruction type	Instruction count	Clock cycle count
Integer arithmetic	45000	1
Data transfer	32000	2
Floating point	15000	2
Control transfer	8000	2

Determine the effective CPI, MIPS rate, and execution time for this program. **[2]**

(ii) A workstation uses a 15-MHZ processor with a claimed 10-MIPS rating to execute a given program mix. Assume a one-cycle delay for each memory access. **[2 + 2]**

(a) What is the effective CPI of this computer?

(b) Suppose the processor is being upgraded with a 30-MHz clock. However, the speed of the memory subsystem remains unchanged, and consequently, two clock cycles are needed per memory access. If 30% of the instructions require one memory access and another 5% require two memory accesses per instruction, what is the performance of the upgraded processor with a compatible instruction set and equal instruction counts in the given program mix?

Q4. Analyze the data dependences among the following statements in a given program: **[2 + 2 + 2]**

S1: Load R1, 1024	/R1 ← 1024/
S2: Load R2, M(10)	/R2 ← Memory(10)/
S3: Add R1, R2	/R1 ← (R1) + (R2)/
S4: Store M(1024), R1	/Memory(1024) ← (R1)/
S5: Store M(R2), 1024	/Memory(64) ← 1024/

where (Ri) means the content of register Ri and Memory(10) contains 64 initially.

(a) Draw a dependence graph to show all the dependences.
(b) Are there any resource dependences if only one copy of each functional unit is available in the CPU?
(c) Repeat the above for the following program statements:

S1: Load R1, M(100)	/R1 ← Memory(100)/
S2: Move R2, R1	/R2 ← (R1)/
S3: Inc R1	/R1 ← (R1) + 1/
S4: Add R2, R1	/R2 ← (R2) + (R1)/
S5: Store M(100), R1	/Memory(100) ← (R1)/

Q5. Consider the execution of the following code segment consisting of seven statements. Use Bernstein's conditions to detect the maximum parallelism embedded in this code. justify the portions that can be executed in parallel and the remaining portions that must be executed sequentially. Rewrite the code using parallel constructs such as Cobegin and Coend. No variable substitution is allowed. All statements can be executed in parallel if they are declared within the same block of a (**Cobegin, Coend**) pair. **[2 + 2 + 2]**

S1: A = B + C	S2: C = D + E	S3: F = G + E	
S4: C = A + F	S5: M = G + C	S6: A = L + C	S7: A = E + A

Q6. Consider the following assembly language code. Exploit the maximum degree of parallelism among the 16 instructions, assuming no resource conflicts and multiple functional units are available simultaneously. For simplicity, no pipelining is assumed. All instructions take one machine cycle to execute. Ignore all other overhead. **[1+2+2+1]**

1: Load R1, A	/R1 ← Mem(A)/
2: Load R2, B	/R2 ← Mem(B)/
3: Mul R3, R1, R2	/R3 ← (R1) x (R2)/
4: Load R4, D	/R4 ← Mem(D)/
5: Mul R5, R1, R4	/R5 ← (R1) x (R4)/
6: Add R6, R3, R5	/R6 ← (R3) + (R5)/
7: Store X, R6	/Mem(X) ← (R6)/
8: Load R7, C	/R7 ← Mem(C)/
9: Mul R8, R7, R4	/R8 ← (R7) x (R4)/
10: Load R9, E	/R9 ← Mem(E)/
11: Add R10, R8, R9	/R10 ← (R8) + (R9)/
12: Store Y, R10	/Mem(Y) ← (R10)/
13: Add R11, R6, R10	/R11 ← (R6) + (R10)/
14: Store U, R11	/Mem(U) ← (R11)/
15: Sub R12, R6, R10	/R12 ← (R6) - (R10)/
16: Store V, R12	/Mem(V) ← (R12)/

(a) Draw a program graph with 16 nodes to show the flow relationships among the 16 instructions.
(b) Consider the use of a three-issue superscalar processor to execute this program fragment in minimum time. The processor can issue one memory-access instruction (Load or Store but not both), one Add/Sub instruction, and one Mul (multiply) instruction per cycle. The Add unit, Load/Store unit, and Multiply unit can be used simultaneously if there is no data dependence.
(c) Repeat part (b) on a dual-processor system with shared memory. Assume that the same superscalar processors are used and that all instructions take one cycle to execute. Partition the given program into two balanced halves. You may insert some load or store instructions to pass intermediate results generated by the two processors to each other. Show the divided program flow graph with the final output U and V generated by the two processors separately.
(d) Work out an optimal schedule for parallel execution of the above (part c) divided program by the two processors in minimum time.