

1

Set 1 ANUSHTHAN SAXENA

S20210010227

$$Q_1 \quad (1432)_{10} = (\quad)_{16} < (\quad)_2 = (\quad)_8$$

16	1432	
16	89	8
16	5	9
	0	*5

$$(1432)_{10} = (\cancel{998})_{16}^{598}$$

For decimal to hexadecimal (or any conversion really), divide the decimal number by the new base, with the remainders forming the new digit this is the easiest method of conversion.

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1
1	0	1	1	0	0	1	1	0	0	0

$$(1432)_{10} = 1 \times (2^{10}) + 0 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3$$

$$\Rightarrow (1432)_{10} = (10110011000)_2$$

8	1432	
8	179	0
8	22	3
8	2	6
	0	2

$$(1432)_{10} = (2630)_8$$

Q2 8 platters, 256 tracks per surface, 512 sectors/track,
512 KB/sector

$$\text{Capacity (in KB)} = 8 \times 2 \times 256 \times 512 \times 512 \\ = 2^4 \times 2^8 \times 2^9 \times 2^9 = 2^{30} \text{ KB}$$

$$\text{Capacity (in TB)} = \frac{2^{30} \text{ KB}}{1024 \times 1024 \times 1024} = \underline{\underline{1 \text{ TB}}}$$

Q3 → Kernel is the core component of an operating system responsible for resource management, interrupt management, file and I/O management and security handling.

→ Kernel consists of these 4 essential components:

- (i) Memory manager.
- (ii) Interrupt handler.
- (iii) I/O manager
- (iv) Scheduler.

→ Being the memory manager of a system, it allocates and de-allocates memory associated with any application the user is using, makes sure there is no accidental overlap of memory sectors for multiple programs.

- Since an operating system is usually interrupt driven, interrupt handling is ~~essentially~~ arguably the most important function of the kernel.
It handles any interrupt and directs the necessary requests and instructions to the CPU rather than the CPU unnecessarily translating these interrupts.
- Also handles every system call whether it's for spawning a new process or requirement of additional resources by an application.
- Handles I/O by communicating with the different device controllers through a device driver in an interrupt-driven process.
- Works with the clock to schedule tasks and resources, essential for proper smooth functioning of any system.
- Also helps with security. Continuing the I/O point, it manages a file system, defining address to various information and facilitates its storage on the secondary memory.

Q1 → Assembler is a tool which converts assembly language instructions to machine language instructions.

Consequently, it is required wherever there are some assembly language instructions involved.

In many ~~compl~~ compiled languages (like C), during the compilation process the instructions are first translated from high-level to ~~low~~ assembly level.

Also, where the speed and size of the program is an essential factor of consideration, the instructions are generally written in assembly language (eg. controllers' or microprocessors) instructions.

Being Assembly language being extensively used everywhere makes assembler an essential programming tool.

Q5 CREATE table Job_application (

NAME var_char(20)

id var char(30)

contact num(10));

Q1

Hard links

- The actual link of a file and its contents.
- Link stays the same even when the attached contents are subjected to some change.

Soft links

- A virtual link, which works like a shortcut, and is not an absolute link.
- ~~Link~~ A new link has to be formed in situations where original file is ~~modified~~ changed.

- For security purposes, the operating system / user can give different files different read, write and executable permissions.

Any file, when listed with its file permissions. Eg-

-r-- rwx rwx file1

~~file~~ "ls -l" shows

The first position is d for a directory, empty for a file.
File permissions go in the order user → group → all.

r: read permission (read from file)

w: write permission (write to file)

x: executable permission (Allows a file to be executed)

→ touch my-file.txt

echo "This is for midsem" > my-file.txt

ls -l

(Output : -rw-r--r-- my-file.txt)
Example

chmod 00-rwx my-file.txt

cat my-file.txt

↳ Will not show the file contents since the file doesn't have read permission.

Q7 #! /bin/bash

read num1 num2 num3

if [\$num1 -gt \$num2] && [\$num1 -gt \$num3]
then \$greatest=\$num1;

elif [\$num2 -gt \$num1] && [\$num2 -gt \$num3]
then \$greatest=\$num2

else \$greatest=\$num3

fi

7

Set 1 S20210010027 ANUSHTHAN SAXENA

if [[\$num1 -lt \$num2] && [\$num1 -lt \$num3]]

then \$smallest = \$num1

elif [[\$num2 -lt \$num1] && [\$num2 -lt \$num3]]

then \$smallest = \$num2

else \$smallest = \$num3

fi

product = \$(expr \$greatest * \$smallest)

temp=\$smallest

rev=0

digit=0

while [\$temp -gt 0]

do

digit = \$(expr \$temp / 10)

rev = \$(expr \$rev * 10 + \$digit)

temp = \$(expr \$temp / 10)

done

diff = \$(expr \$smallest - \$rev)