thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.

NAME - Akanksha Saxena Date -

COURSE CODE - MCS-041 SEMESTER - 4th

COURSE TITLE : OPERATING SYSTEMS

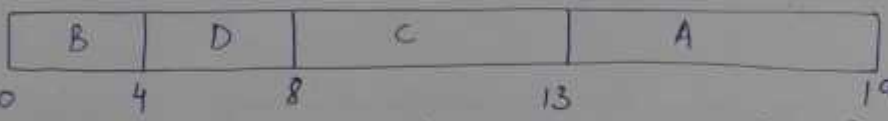
ROLL NO. -

Ques 1

Solution :

(a) Using the Shortest-Job-First method (SJF)

Using the SJF scheduling scheme because the shortest length of process will first get execution, the Gantt chart will be



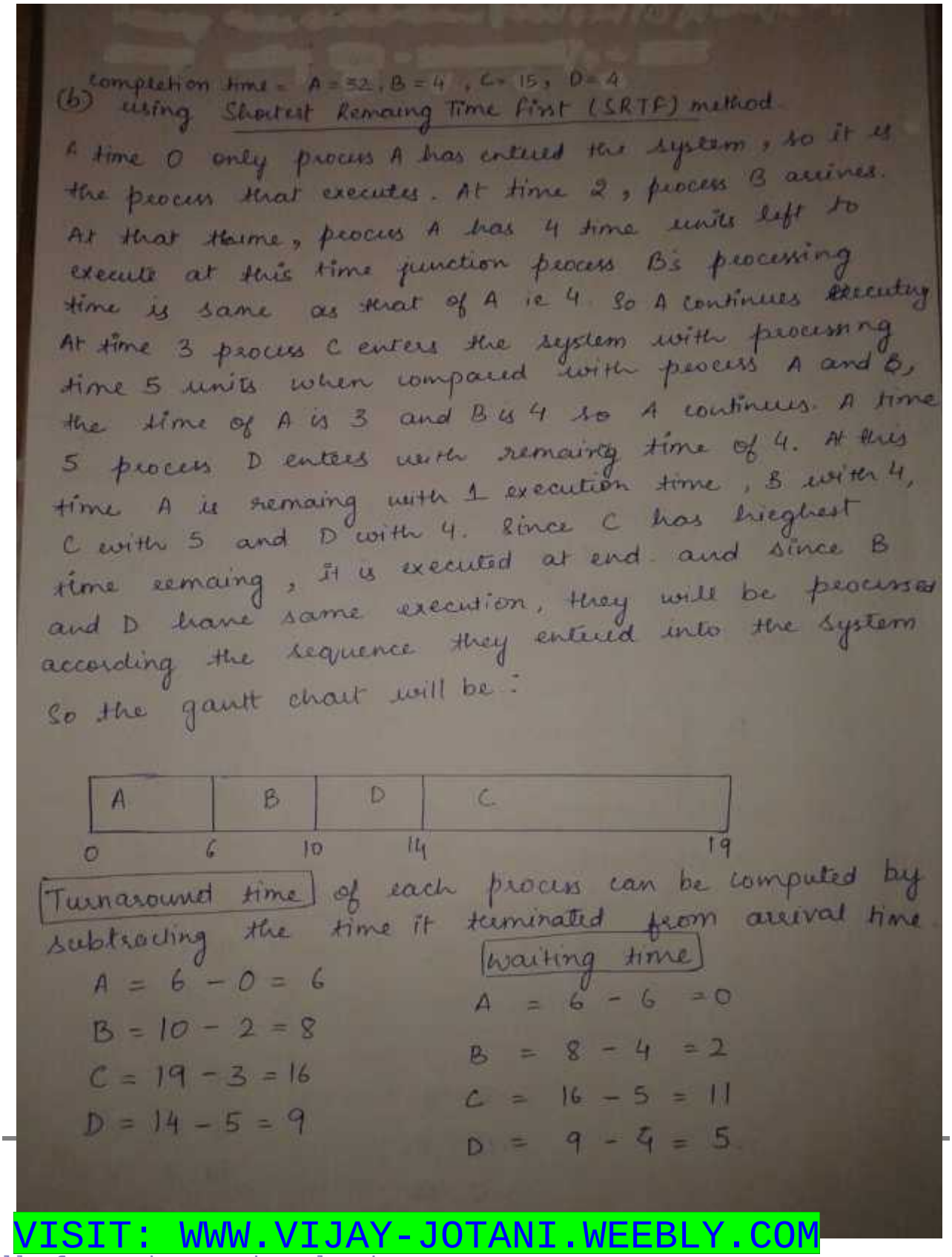
Because shortest running time is of the process B, then process D and then C and then A. The waiting time for process A is 13 ms, for process B is 0 ms, for process C is 8ms and for process D is 4ms. as -

Time	Process Completed	Turn Around Time	Wating Time
0	-	-	-
4	B	4 - 2 = 2	4 - 2 = 2
8	D	8 - 5 = 3	4 - 3 = 1
13	C	13 - 3 = 10	10 - 5 = 5
19	A	19 - 0 = 19	19 - 6 = 13

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.



(b) completion time - A = 32, B = 4, C = 15, D = 4
using Shortest Remaing Time First (SRTF) method

At time 0 only process A has entered the system, so it is the process that executes. At time 2, process B arrives. At that time, process A has 4 time units left to execute at this time junction process B's processing time is same as that of A ie 4. So A continues executing At time 3 process C enters the system with processing time 5 units when compared with process A and B, the time of A is 3 and B is 4 so A continues. A time 5 process D enters with remaing time of 4. At this time A is remaing with 1 execution time, B with 4, C with 5 and D with 4. Since C has hieghest time remaing, it is executed at end and since B and D have same execution, they will be processed according the sequence they entered into the system

So the gantt chart will be:

A	B	D	C
0	6	10	14	19

Turnaround time of each process can be computed by subtracting the time it taminated from arrival time.

A = 6 - 0 = 6
B = 10 - 2 = 8
C = 19 - 3 = 16
D = 14 - 5 = 9

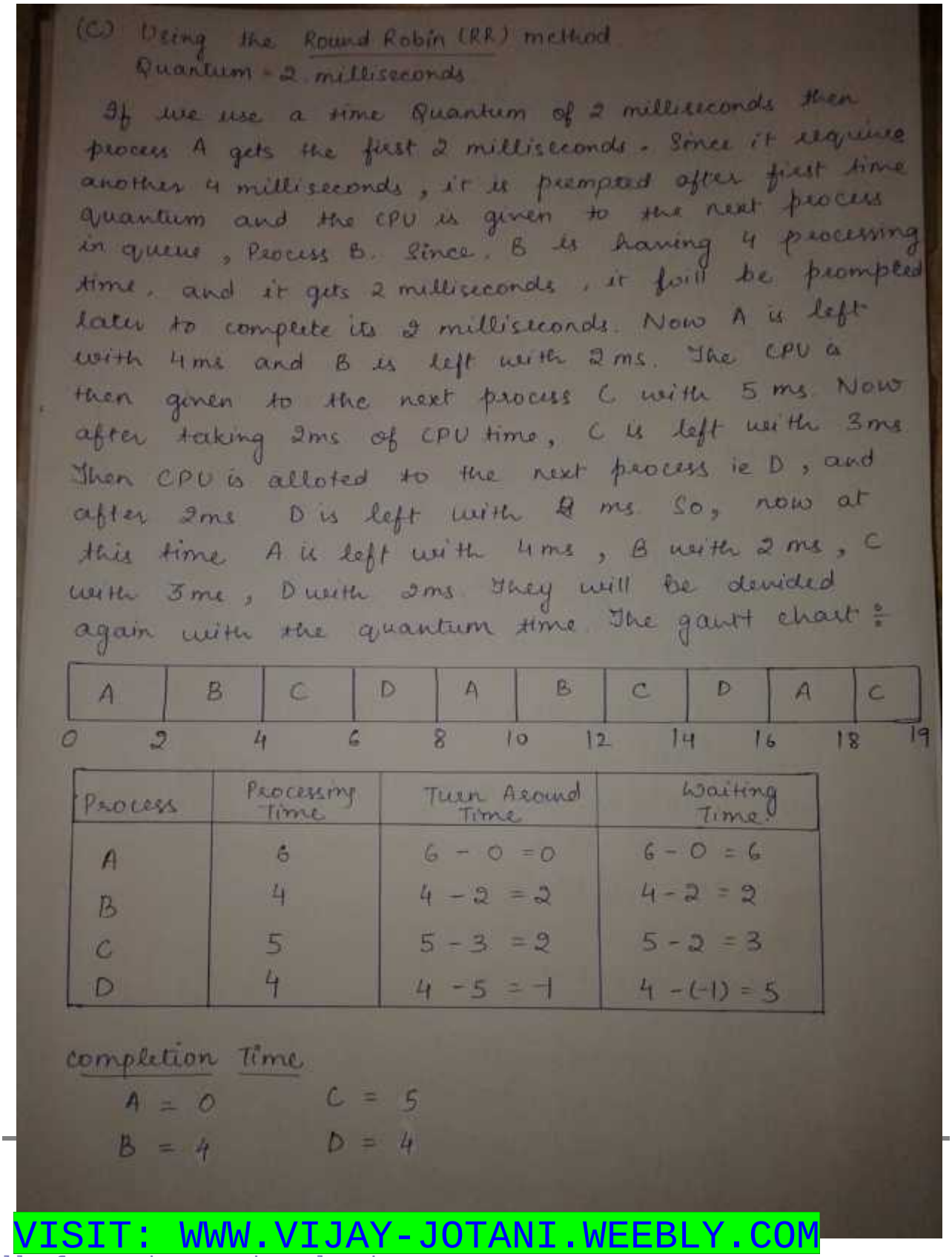
Waiting time

A = 6 - 6 = 0
B = 8 - 4 = 2
C = 16 - 5 = 11
D = 9 - 4 = 5

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.



(c) Using the Round Robin (RR) method
Quantum = 2 milliseconds

If we use a time Quantum of 2 milliseconds then process A gets the first 2 milliseconds. Since it require another 4 milliseconds, it is prempted after first time quantum and the CPU is given to the next process in queue, Process B. Since, B is having 4 processing time, and it gets 2 milliseconds, it will be prompted later to complete its 2 milliseconds. Now A is left with 4ms and B is left with 2ms. The CPU is then given to the next process C with 5 ms. Now after taking 2ms of CPU time, C is left with 3ms. Then CPU is alloted to the next process ie D, and after 2ms D is left with 2 ms. So, now at this time A is left with 4ms, B with 2 ms, C with 3ms, D with 2ms. They will be devided again with the quantum time. The gantt chart :-

A	B	C	D	A	B	C	D	A	C

0 2 4 6 8 10 12 14 16 18 19

Process	Processing Time	Turn Around Time	Waiting Time
A	6	6 - 0 = 0	6 - 0 = 6
B	4	4 - 2 = 2	4 - 2 = 2
C	5	5 - 3 = 2	5 - 2 = 3
D	4	4 - 5 = -1	4 - (-1) = 5

Completion Time

A = 0 C = 5

B = 4 D = 4

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.

Ques 2

Solution Bankers Algorithm

```
# include <stdio.h>
# include <conio.h>
void main () {
    int k=0, output[10], d=0, t=0, ins[5], i;
    int avail[5], allocated[10][5], need[10][5], MAX[10][5];
    int pno, P[10], j, rz, count=0;
    clrscr();
    printf ("\n Enter the number of resources");
    scanf ("%d", &rz);
    printf ("\n Enter the max instance of each resources");
    for (i=0; i<rz; i++) {
        avail[i] =0;
        printf ("%c=", (i+97));
        scanf ("%d", &ins[i]);
    }
    printf ("\n Enter number of processes");
    scanf ("%d", &pno);
    printf ("\n Enter the allocation matrix \n");
    for (i=0; i<rz; i++)
    printf ("%c", (i+97));
    printf ("\n");
    for (i=0; i<pno; i++) {
        P[i] = i;
        printf ("P[%d] ", P[i]);
        for (j=0; j<rz; j++) {
            scanf ("%d", &allocated[i][j]);
            avail[j] += allocated[i][j];
```


Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.

```
printf("\n Enter the MAX matrix \n");
for (i=0;i<r2;i++){
    printf("%c", (i+97));
    avail[i] = ins[i] - avail[i];
}
printf("\n");
for (i=0;i<pno;i++) {
    printf("P[%d] ", i);
    for (j=0;j<r2;j++)
    scanf("%d", &MAX[i][j]);
}
print("\n");
A: d=-1;
for (i=0;i<pno;i++) {
   count=0;
   t=P[i];
   for(j=0;j<r2;j++){
      need[t][j]=MAX[t][j] - allocated[t][j];
      if (need[t][j] <= avail[j])
         count++;
   }
   if (count==r2) {
      output[k++] = P[i];
      for (j=0;j<r2;j++)
      avail[j] += allocated[t][j];
   } else
     P[++d] = P[i];
} if (d!=-1) {
      pno = d+1;
      goto A; }
printf("\t <");
```


Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.

```
for (i=0 ; i<k;i++)
 printf ("*");
  getch();
}
```

(b)

Solution

(i) Least Recently Used (LRU) Algorithm

Page-referencing string

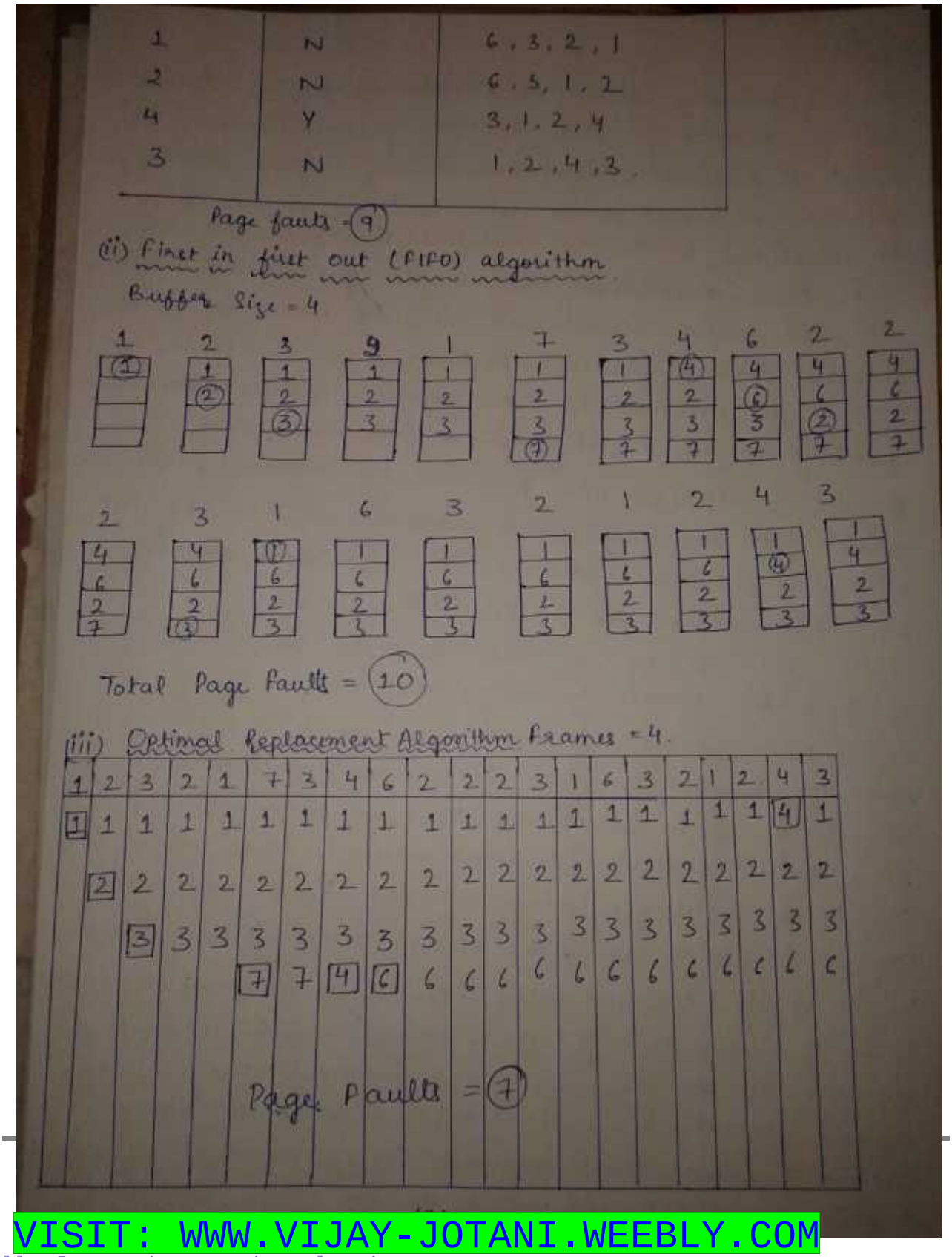
1, 2, 3, 2, 1, 7, 3, 4, 6, 2, 2, 2, 3, 1, 6, 3, 2, 1, 2, 4, 3

Page Refrd.	Page Fault	Resulting list (frame = 4)
1	Y	1
2	Y	1, 2
3	Y	1, 2, 3
2	N	1, 3, 2
1	N	3, 2, 1
7	Y	3, 2, 1, 7
3	N	2, 1, 7, 3
4	Y	1, 7, 3, 4
6	Y	7, 3, 4, 6
2	Y	3, 4, 6, 2
2	N	3, 4, 6, 2
2	N	3, 4, 6, 2
3	N	4, 6, 2, 3
1	Y	6, 2, 3, 1
6	N	2, 3, 1, 6
3	N	2, 1, 6, 3
2	N	1, 6, 3, 2

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.



1	N	6, 3, 2, 1
2	N	6, 5, 1, 2
4	Y	3, 1, 2, 4
3	N	1, 2, 4, 3

Page faults = 9

(ii) First in first out (FIFO) algorithm

Buffer Size = 4

1 2 3 9 1 7 3 4 6 2 2

2 3 1 6 3 2 1 2 4 3

Total Page Faults = 10

(iii) Optimal Replacement Algorithm Frames = 4

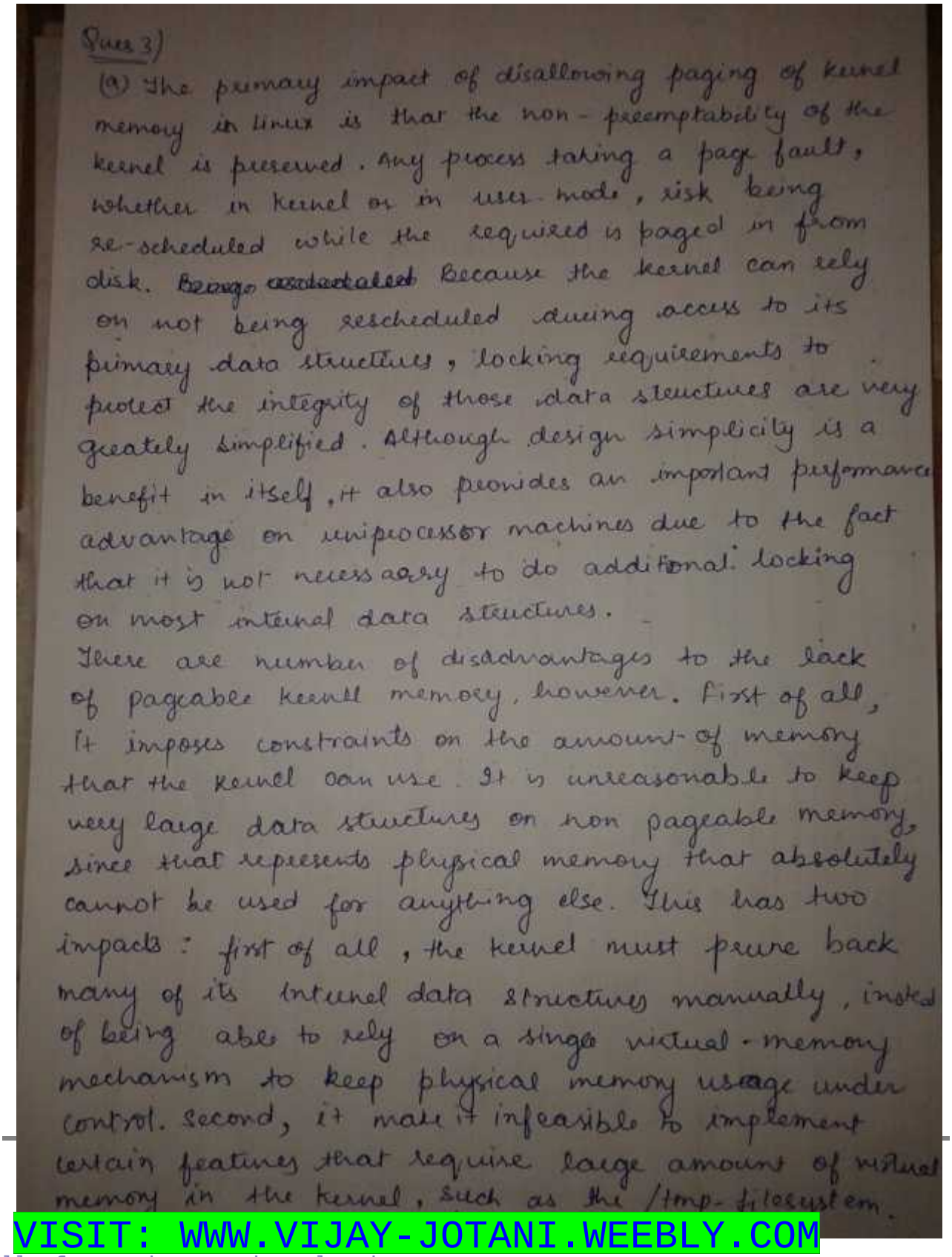
1	2	3	2	1	7	3	4	6	2	2	2	3	1	6	3	2	1	2	4	3
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	1
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
					7	7	4	6	6	6	6	6	6	6	6	6	6	6	6	6

Page Faults = 7

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.



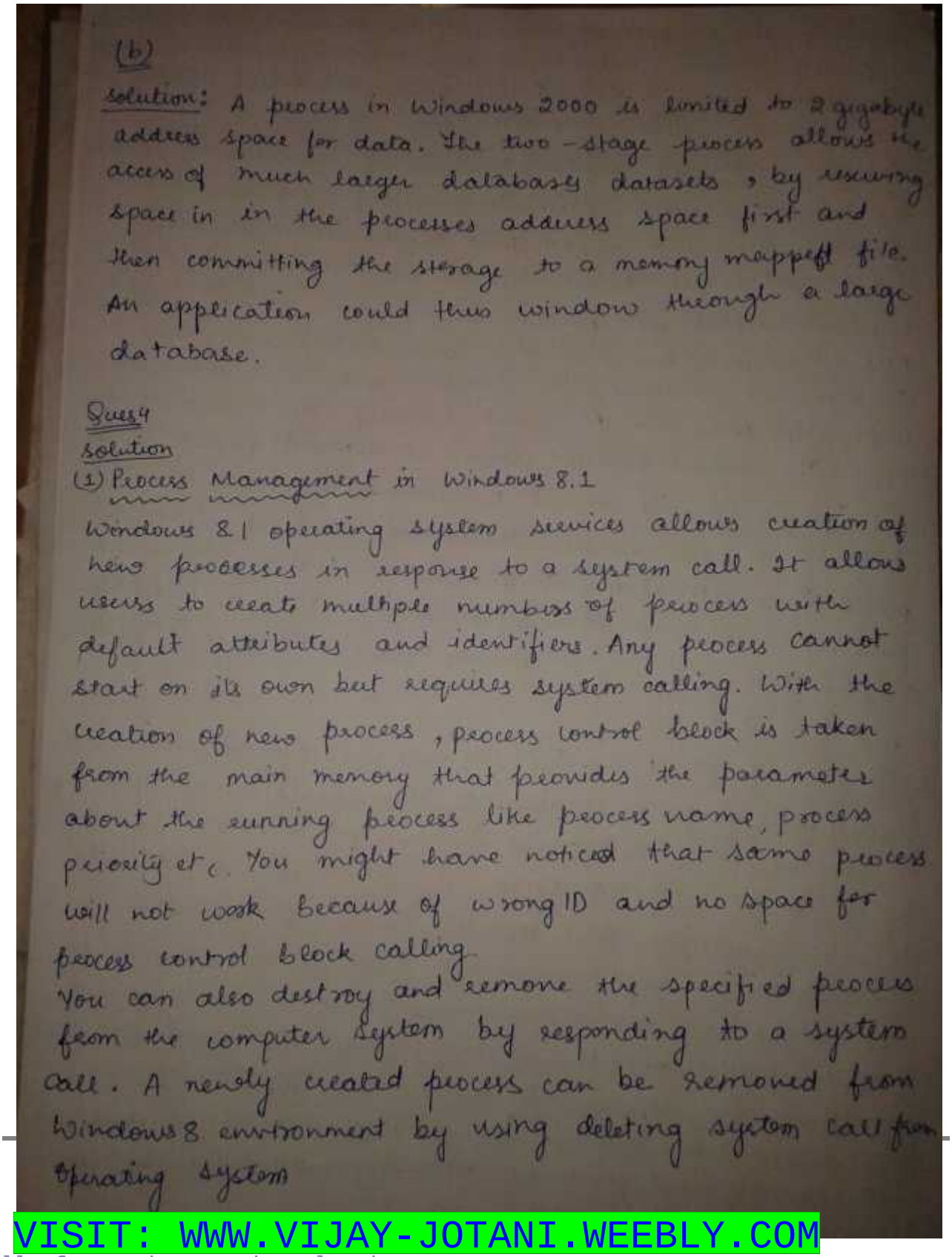
Ques 3)

(a) The primary impact of disallowing paging of kernel memory in Linux is that the non-preemptability of the kernel is preserved. Any process taking a page fault, whether in kernel or in user mode, risk being re-scheduled while the required is paged in from disk. ~~Because~~ ~~rescheduled~~ Because the kernel can rely on not being rescheduled during access to its primary data structures, locking requirements to protect the integrity of those data structures are very greately simplified. Although design simplicity is a benefit in itself, it also provides an important performance advantage on uniprocessor machines due to the fact that it is not necessary to do additional locking on most internal data structures.

There are number of disadvantages to the lack of pageable kernel memory, however. First of all, it imposes constraints on the amount of memory that the kernel can use. It is unreasonable to keep very large data structures on non pageable memory, since that represents physical memory that absolutely cannot be used for anything else. This has two impacts: first of all, the kernel must prune back many of its internal data structures manually, instead of being able to rely on a single virtual-memory mechanism to keep physical memory usage under control. Second, it make it infeasible to implement certain features that require large amount of virtual memory in the kernel, such as the /tmp filesystem.

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.



(b)

Solution: A process in Windows 2000 is limited to 2 gigabyte address space for data. The two-stage process allows the access of much larger database datasets, by reserving space in the processes address space first and then committing the storage to a memory mapped file. An application could thus window through a large database.

Ques4

solution

(1) Process Management in Windows 8.1

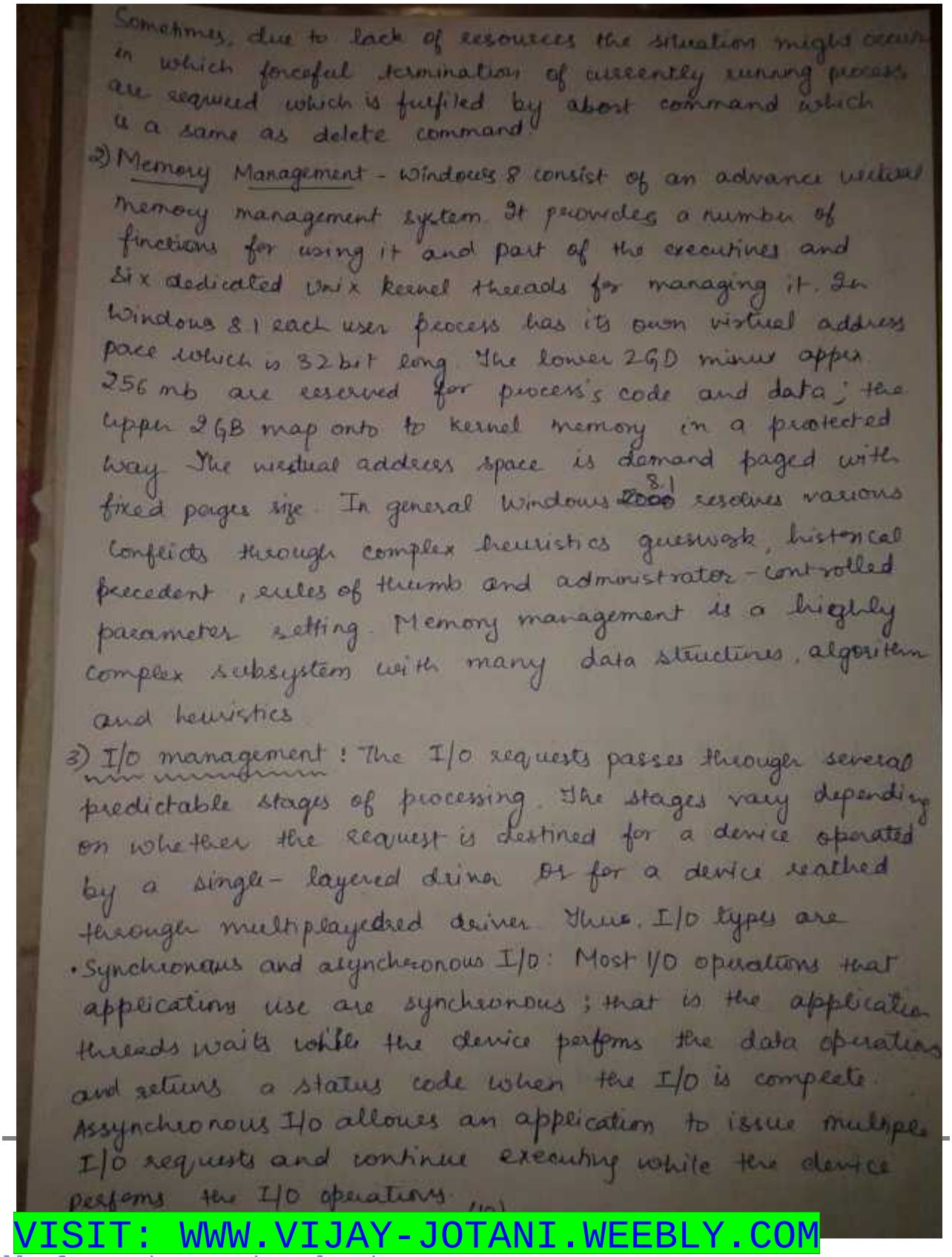
Windows 8.1 operating system services allows creation of new processes in response to a system call. It allows users to creat multiple numbers of process with default attributes and identifiers. Any process cannot start on its own but requires system calling. With the creation of new process, process control block is taken from the main memory that provides the parameters about the running process like process name, process priority etc. You might have noticed that same process will not work because of wrong ID and no space for process control block calling.

You can also destroy and remove the specified process from the computer system by responding to a system call. A newly created process can be removed from Windows 8 environment by using deleting system call from Operating system

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.



Sometimes, due to lack of resources the situation might occur in which forceful termination of currently running process are required which is fulfiled by abort command which is a same as delete command.

2) Memory Management - Windows 8 consist of an advance virtual memory management system. It provides a number of functions for using it and part of the executives and six dedicated Unix kernel threads for managing it. In Windows 8.1 each user process has its own virtual address pace which is 32 bit long. The lower 2GD minus appx 256 mb are reserved for process's code and data; the upper 2GB map onto to kernel memory in a protected way. The virtual address space is demand paged with fixed pages size. In general Windows ~~2000~~ 8.1 resolves various conflicts through complex heuristics guesswork, historical precedent, rules of thumb and administrator-controlled parameter setting. Memory management is a highly complex subsystem with many data structures, algorithm and heuristics

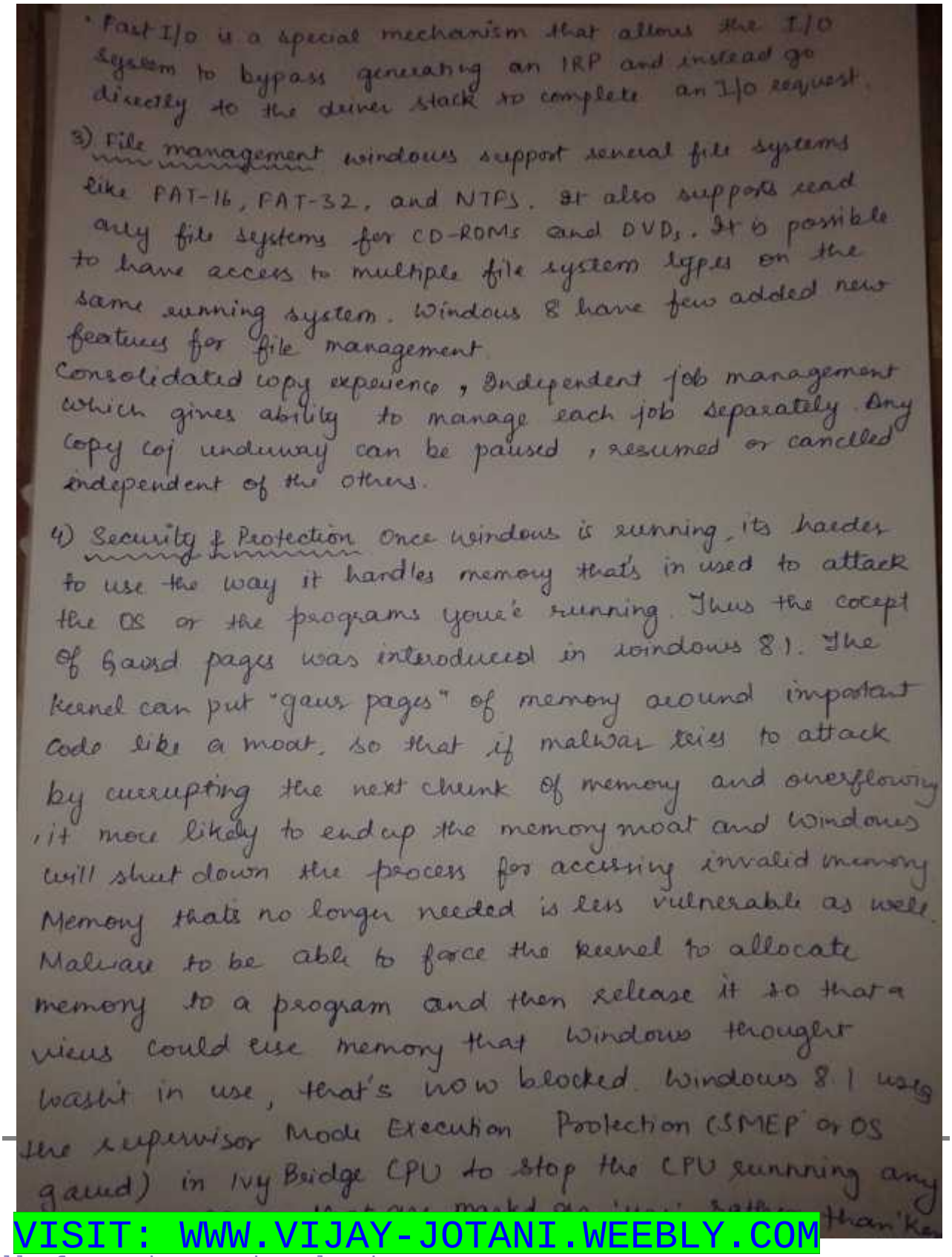
3) I/O management : The I/O requests passes through several predictable stages of processing. The stages vary depending on whether the request is destined for a device operated by a single-layered driver or for a device reached through multiplayered driver. There I/O types are

- Synchronous and asynchronous I/O: Most I/O operations that applications use are synchronous; that is the application threads waits while the device performs the data operations and returns a status code when the I/O is complete. Assynchronous I/O allowes an application to issue multiple I/O requests and continue executing while the device performs the I/O operations.

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)

thanx to Akanksha

Hi everyone, I am ***Akanksha***. I am not sure if all the answers are corr
this is all it could come up with. Please read carefully before writing, I am not
responsible for anything thing. Thank you all.



• Fast I/O is a special mechanism that allows the I/O system to bypass generating an IRP and instead go directly to the driver stack to complete an I/O request.

3) File management windows support several file systems like FAT-16, FAT-32, and NTFS. It also supports read only file systems for CD-ROMs and DVDs. It is possible to have access to multiple file system types on the same running system. Windows 8 have few added new features for file management.
Consolidated copy experience, Independent job management which gives ability to manage each job separately. Any copy or move underway can be paused, resumed or cancelled independent of the others.

4) Security & Protection Once windows is running, its harder to use the way it handles memory that's in used to attack the OS or the programs you're running. Thus the concept of Guard pages was introduced in windows 8.1. The kernel can put "guard pages" of memory around important code like a moat, so that if malware tries to attack by corrupting the next chunk of memory and overflowing, it more likely to end up the memory moat and Windows will shut down the process for accessing invalid memory. Memory that's no longer needed is less vulnerable as well. Malware to be able to force the kernel to allocate memory to a program and then release it so that a virus could use memory that Windows thought wasn't in use, that's now blocked. Windows 8.1 uses the supervisor Mode Execution Protection (SMEP or OS guard) in Ivy Bridge CPU to stop the CPU running any

Thanx to all of you, i m now in malaysia..
btw i have a question "would u like to be a friendship with me?"
yaar bahut lonely ho gya hu guyz :):)