**Q4**

The chunking is a process by which individual pieces of an information set are broken down and then grouped together in a meaningful whole, so The chunks by which the information is grouped is meant to improve short-term retention of the material, thus bypassing and seperating the limited capacity of working memory

**A**

If you want to see your cached memory of your processor . then you have to go to the BIOS Setup . which is normally F10 in most of the computers . or may be F4 or F12.

you have to press the key on start-up of your computer. then you have to go on system information.

SO THE PROCESS IS THE FOLLOWING :

1. press the start button of your CPU . or just restart you computer.

2. press f10 (for hp) and f9 , f12 for others. press key for 2 to 3 times.

3.then go to the system information. and press enter . (note that: your mouse will not work in bios menu .)

**B**

Virtual addresses are used by the operating system to access kernel and user memory. The CPU manages translation of virtual to physical addresses using its Memory Management Unit (MMU). A virtual address is specified as a offset from the start of a memory segment; these segments are used by the kernel and user processes to hold their text, stack, data, and other regions.

step 1

0x000039A is in virtual, lets convert it to binary representation

0b 0000 0000 0000 0000 0011 1001 1010

0x 0 0 0 0 3 9 A

step2

Calculate the number of bits needed to reference the whole 1KB

1K = 2^10

==> 10 bits are needed. Just do log2(page-size).

step3

Take away the first 10 bits of the binary presentation

0b 0000 0000 0000 0000 0011 1001 1010

offset = 0b 11 1001 1010

= 0x 3 9 A

step4

Get the virtual page out of what ever bits left

0b (00)(00 00)(00 00)(00 00)(00 00)

= 0x00000

step5

Go to the page table at the entry 0x00000, there you will find the corresponding frame number.

Suppose the page table is given:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 0x0 | 0x0100 |

| 0x1 | 0xA |

| . | |

| . | |

| | |

----------------

step6

Turn the frame number to binary representation and concatenate it to the offset

Frame | offset

0x0100 |

0b (00)(00 00)(01 00)(00 00)(00 | 11) (1001) (1010)

0x 004039A

**Q5**

#include "idnm.h"

int main(int argc, char \*argv[]){

//(A)

if(argc<2) {

cout<<"arguments are invalid!"<<endl;

return -1;

//(B)

int search\_id=atoi(argv[1]);

//set up for memory mapped file

int fd=fopen(filename, O\_RDONLY)

struct stat stat\_buf;

fstat(fd, &stat\_buf)

int size=stat\_buf.st\_size;

//(C)

void \*file\_bytes=stat\_buf.address

//(D)

fhead\_t \*fhead=(fhead\_t\*)malloc(sizeof(fhead\_t));

//(E)

idmn\_t \*idmn\_sec=(idmn\_t\*)malloc(sizeof(idmn\_t));

char \*nm\_sec=fhead.nm\_sec\_off;

//(F)

char \*name=fhead->nm\_off;

//(G)

if(fhead->id==i){

printf("%x\n",\*fhead->id);

printf("%s\n",\*fhead->name);

}

}