TD 6 Operating Systems

Problem 2) What is the difference between a physical address and a virtual address ?

A physical adress refers to aa hardware adress of physical memory and a virtual adresse refers to the virtual store viewed by the process.

Problem 3) Explain under which circumstances page faults occur and how they are resolved by the operating system

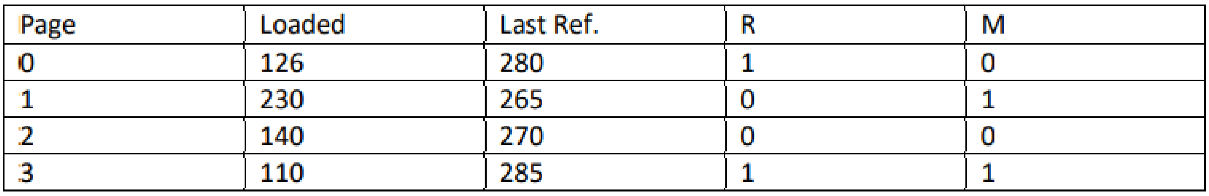
A page fault occurs when a program tries to access a block of memory that is not stored in the physical memory, or RAM.

Then, it tries to find out which virtual page is needed, then the page is scheduled for transfer to disk, and the fault is suspended. Another process is made to run until disk transfer is completed.

Problem 4) Explain the purpose of pages and frames as they relate to executing processes.

Page tables allow a process to use non-contiguous blocks of physical memory by mapping page of logical memory to frame of physical memory through the use of a page table.

Problem 5) A computer has four frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the time are in clock ticks) :



Page	Loaded	Last Ref.	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

(a) Which page will NRU replace?

NRU will replace page 2 because R = 0 and M = 0.

(b) Which page will FIFO replace?

FIFO will replace page 3 because it is the oldest loaded page.

(c) Which page will LRU replace?

LRU will replace page 1 because it is the oldest referenced time.

Problem 6) If FIFO page replacement is used with four page frames and eight pages, how many page faults will occur with the reference string 0172327103 if the four frames are initially empty?

There will be 6 page faults for the first 4 numbers, the first “3” and the second “0”.

Problem 7) If LRU page replacement is used with four page frames and eight pages, how many page faults will occur with the reference string 0172327103 if the four frames are initially empty?

There are 7 page faults for the first 4 numbers, the first “3”, the second “0” and the last number : “3”.

Problem 8) Using the page table below, give the physical address corresponding to each of the following virtual addresses:

(a) 20

0<20<4000 → page frame 2 : 12K = 12*1024 = 12288

12288 + 20 = 12308

(b) 8200

8000<8200<12000 → page frame 6 : 28K = 28*1024 = 28672

28672 + 8200 = 36 872

(c) 16536

16000<16536<20000 → page frame 4 : 20K = 20*1024 = 20480

20480 + 16536 = 37016

Problem 9) Consider a swapping system in which memory consists of the following hole sizes in memory order: 10 MB, 4MB, 20 MB, 18 MB, 7 MB, 9 MB, 12 MB and 15 MB. Which hole is taken for successive segment of

(a) 12 MB

(b) 10 MB

(c) 9 MB

for first fit? Now repeat the question for best fit, worst fit, and next fit

first fit : 20MB 10MB 18MB

best fit : 12MB 10MB 9MB

worst fit : 20MB 18MB 15MB

next fit : 20MB 18MB 9MB

Problem 10) A computer system has a 32-bit virtual address space with a page size of 8K, and 4 bytes per page table entry

(a) How many pages are in the virtual address space?

8K = 8*1024 = 8*2^10 = 2^13

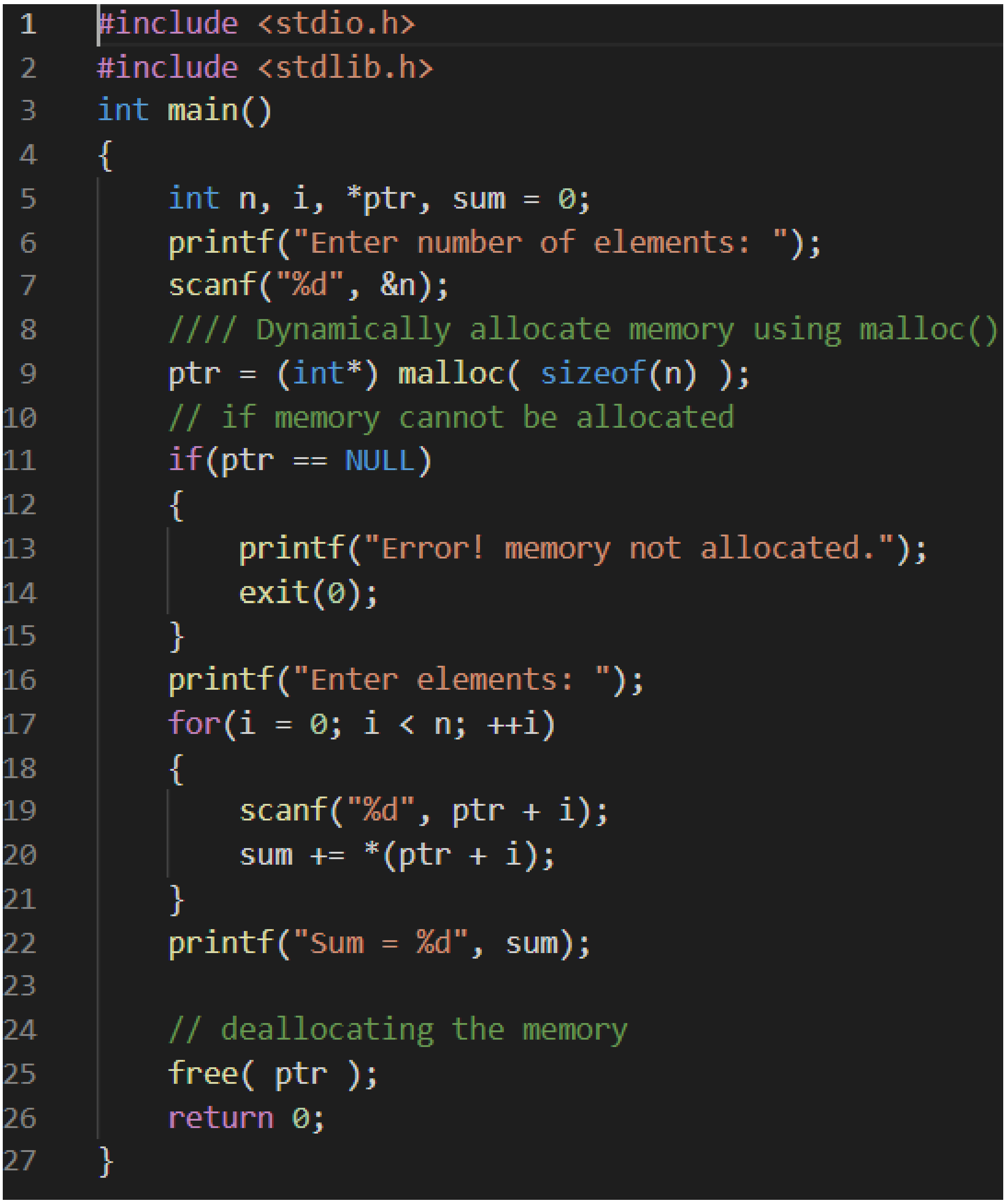
2^32/2^13 = 2^19

There are 2^19 pages in the virtual address space.

(b) What is the maximum size of addressable physical memory in this system?

The maximum size of addressable physical memory in this system is 2^32*2^13 = 2^45.

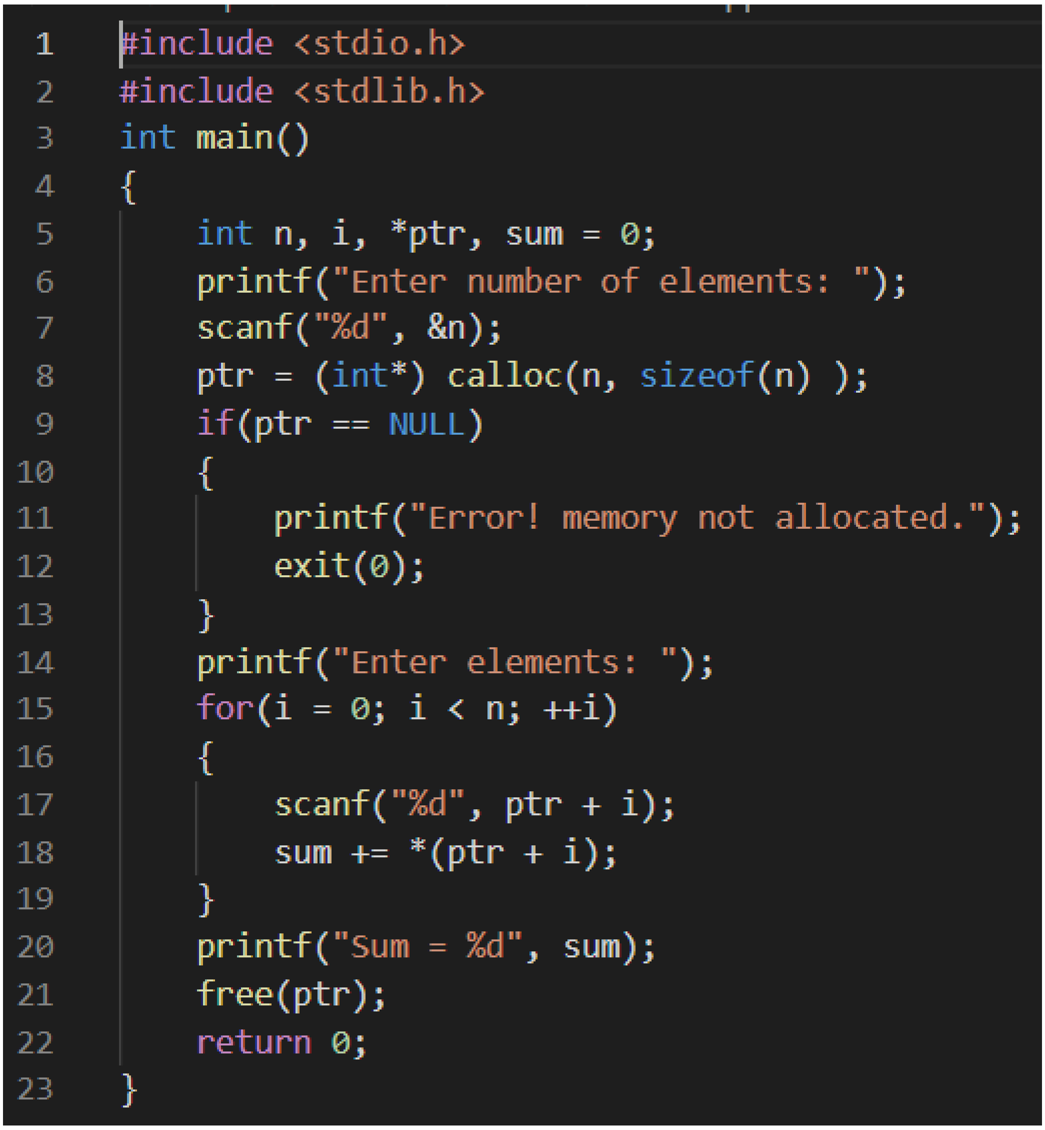
Problem 11) The code C below dynamically allocated the memory for n number of type int. Analyze this code and complete the parts indicated in red in it.



```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    //// Dynamically allocate memory using malloc()
    ptr = (int*) malloc( sizeof(n) );
    // if memory cannot be allocated
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
    printf("Sum = %d", sum);

    // deallocating the memory
    free( ptr );
    return 0;
}
```

Problem 12) The code C below dynamically allocated to calculate the sum of n numbers entered by the user the. Analyze this code and complete the parts indicated in red in it.



```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    ptr = (int*) calloc(n, sizeof(n) );
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
    printf("Sum = %d", sum);
    free(ptr);
    return 0;
}
```