OS: Unit 3 Programming Exercise

PES1UG21CS361
Nandan N
Roll no: 41
F sec

Code:

```c
#include <stdio.h>
struct page
{
    int frame_no;
    int valid;
} typedef Page;
int main()
{
    int page_size = 8;
    int memory_size = 64;
    int max_pages = 8; // Total no of frames
    int no_of_pages;
    int offset = 378;
    printf("Enter the number of process required for the process:");
    scanf("%d", &no_of_pages);
    Page p[no_of_pages];
    if (no_of_pages > max_pages)
        printf("Full memory");

    for (int i = 0; i < no_of_pages; i++)
    {
        printf("For page %d enter frame number:", i);
        scanf("%d", &p[i].frame_no);
        printf("For page %d enter valid bit:", i);
        scanf("%d", &p[i].valid);
    }
    printf("Page table and physcial address\n");
    for (int i = 0; i < no_of_pages; i++)
    {
        printf("%d %d %d \n", i, p[i].frame_no, p[i].valid);
    }
    int page_no;
    printf("page no and offset:");
    scanf("%d,%d", &page_no, &offset);

    printf("frame no, offset:%d,%d", p[page_no].frame_no, offset);
}
```

Output:

```
Enter the number of process required for the process:3
For page 0 enter frame number3
For page 0 enter valid bit1
For page 0 enter frame number:3
For page 0 enter valid bit:1
For page 1 enter frame number:5
For page 1 enter valid bit:1
For page 2 enter frame number:7
For page 2 enter valid bit:1
Page table and physcial address
0 3 1
1 5 1
2 7 1
page no and offset:2,380
frame no, offset:7,380
PS C:\Users\nkuch\OneDrive\Desktop\OS>
```

OS ASSIGNMENT 4

Programming Exercise 4

Write a C program to list all files whose name matches the filter. Inputs to the program as run time arguments: directory and filename (need to support wildcard)

Example: a.out /home/Ubuntu/abc1.txt

Example: a.out /home/Ubuntu/abc*.txt

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<dirent.h>
#include<string.h>
#include<fnmatch.h>

int main(int argc,char *argv[])
{
	DIR *dir;
	struct dirent *entry;
	char *dir_path, *filter;
	size_t dir_len, filter_len;
	if(argc != 3)
	{
		printf("Usage : %s directory filter\n",argv[0]);
		return 1;
	}

	dir_path = argv[1];
	filter = argv[2];
	dir_len = strlen(filter);

	if((dir = opendir(dir_path)) == NULL)
```

```
        {
                printf("Error opening directory %s\n",dir_path);
                return 1;
        }

        while((entry = readdir(dir)) != NULL)
        {
                if(fnmatch(filter,entry->d_name,0)){
                printf("%s/%s\n",dir_path,entry->d_name);
        }


        closedir(dir);
        return 0;
}
}
```

Output: