Question -> Priority Queue Scheduling Algorithm

Garv nanwani

19BCS049

Code -

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct node {
    int data;
    int priority;
    struct node* next;
} Node;
Node* newNode(int d, int p)
{
    Node* temp = (Node*)malloc(sizeof(Node));
    temp->data = d;
    temp->priority = p;
    temp->next = NULL;
    return temp;
}

void traversal(Node** head)
{
    Node* ptr = *head;
    while (ptr != NULL)
    {
        printf("| p%d | %d |  --> ", ptr->data, ptr->priority);
        ptr = ptr->next;
    }
    printf("\n");
}
void pop(Node** head)
{
    if(*head==NULL)
    printf("No process pending\n");
    else{
    Node* temp = *head;
    (*head) = (*head)->next;
    free(temp);
    }
}
void push(Node** head, int d, int p)
{
    Node* start = (*head);
    Node* temp = newNode(d, p);
    if ((*head)->priority > p) {
        temp->next = *head;
        (*head) = temp;
    }
```

```
    else {
        while (start->next != NULL &&
            start->next->priority < p) {
            start = start->next;
        }
        temp->next = start->next;
        start->next = temp;
    }
}
int main()
{
    int x;
    int p,d=0;
    printf("Enter the first process data\n");
    scanf("%d",&d);
    printf("Enter the first process priority\n");
    scanf("%d",&p);
    Node* pq = newNode(d,p);
    int num = 1;
    while(num>0)
    {
        printf("Enter 1 to enter a process\n");
        printf("Enter 2 to execute a process\n");
        printf("Enter 3 to disply all the process\n");
        printf("Enter 4 to exit\n");
        printf("Enter your choice\n");
        scanf("%d",&x);
        switch(x)
        {
            case 1:
            printf("Enter the process data\n");
            scanf("%d",&d);
            printf("Enter the process priority\n");
            scanf("%d",&p);
            push(&pq, d, p);
            break;

            case 2: pop(&pq);
            break;

            case 3: traversal(&pq);
            break;

            case 4: num =0;
            break;

            default: printf("Choice other than 1, 2 and 3\n");
            break;
        }


    }
```

```
    return 0;
}
```

OUTPUT -

```
Enter the first process data
4
Enter the first process priority
5
Enter 1 to enter a process
Enter 2 to execute a process
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
1
Enter the process data
7
Enter the process priority
1
Enter 1 to enter a process
Enter 2 to execute a process
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
1
Enter the process data
3
Enter the process priority
2
Enter 1 to enter a process
Enter 2 to execute a process
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
3
| p7 | 1 |  --> | p3 | 2 |  --> | p4 | 5 |  -->
Enter 1 to enter a process
Enter 2 to execute a process
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
2
Enter 1 to enter a process
Enter 2 to execute a process
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
3
| p3 | 2 |  --> | p4 | 5 |  -->
Enter 1 to enter a process
Enter 2 to execute a process
```

```
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
2
Enter 1 to enter a process
Enter 2 to execute a process
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
3
| p4 | 5 |  -->
Enter 1 to enter a process
Enter 2 to execute a process
Enter 3 to disply all the process
Enter 4 to exit
Enter your choice
4
```