

Ex. No.: 11c)

Date: 18-4-25

Optimal

Aim:

To write a c program to implement Optimal page replacement algorithm.

#### ALGORITHM:

1. Start the process
2. Declare the size
3. Get the number of pages to be inserted
4. Get the value
5. Declare counter and stack
6. Select the least frequently used page by counter value
7. Stack them according the selection.
8. Display the values
9. Stop the process

PROGRAM:

```
#include <stdio.h>
int main() {
    int frames, pages, i, k, page-faults = 0, ref-str[100];
    int memory[10], future[10], flag1, flag2, pos;
    printf("Enter no. of frames: ");
    scanf("%d", &frames);
    printf("Enter no. of pages: ");
    scanf("%d", &pages);
    printf("Enter reference string: ");
    for(i=0; i<pages; i++) {
        scanf("%d", &ref-str[i]);
    }
    for(i=0; i<frames; i++)
        memory[i] = -1;
    printf("In");
    for(i=0; i<pages; i++) {
        flag1 = flag2 = 0;
```

```
for (i=0; i<frames; i++) {
```

```
    if (memory[i] == ref - str[i]) {
```

```
        flag1 = flag2 = 1;
```

```
        break;
```

```
}
```

```
} if (flag1 == 0) {
```

```
    for (i=0; i<frames; i++) {
```

```
        if (memory[i] == ref - str[i]) {
```

```
            flag1 = flag2 = 1;
```

```
            break;
```

```
}
```

```
} if (flag1 == 0) {
```

```
    for (i=0; i<frames; i++) {
```

```
        if (memory[i] == -1) {
```

```
            memory[i] = ref - str[i];
```

```
            page_faults++;
```

```
            flag2 = 1;
```

```
            break;
```

```
}
```

```
}
```

```
} if (flag2 == 0) {
```

```
    for (i=0; i<frames; i++) {
```

```
        future[i] = -1;
```

```
        for (k = i+1; k < pages; k++) {
```

```
            if (memory[k] == ref - str[k]) {
```

```
                future[i] = k;
```

```
                break;
```

```
}
```

```
}
```

```
int max = -1;
```

```
for (i=0; i<frames; i++) {
```

```

if (future[j] == -1) {
    pos = j;
    break;
}

if (future[0] > max) {
    max = future[j];
    pos = j;
}

memory[pos] = ref & fr[i];
page-faults++;

for (j=0; j < frames; j++) {
    if (memory[j] == -1)
        printf(" - ");
    else
        printf("-.%d", memory[j]);
}

printf("\n");
printf("total page faults: %d\n", page-faults);
return 0;
}

```

Enter no. of frames : 2

Enter no. of pages : 6

Enter reference string: 1 5 6 4 5 5

1 -

1 5

6 5

4 5

4 5

4 5

total page faults = 4

**Output:**

*Alt*

**Result:**

Thus the program to implement optimal page replacement technique has been executed successfully.