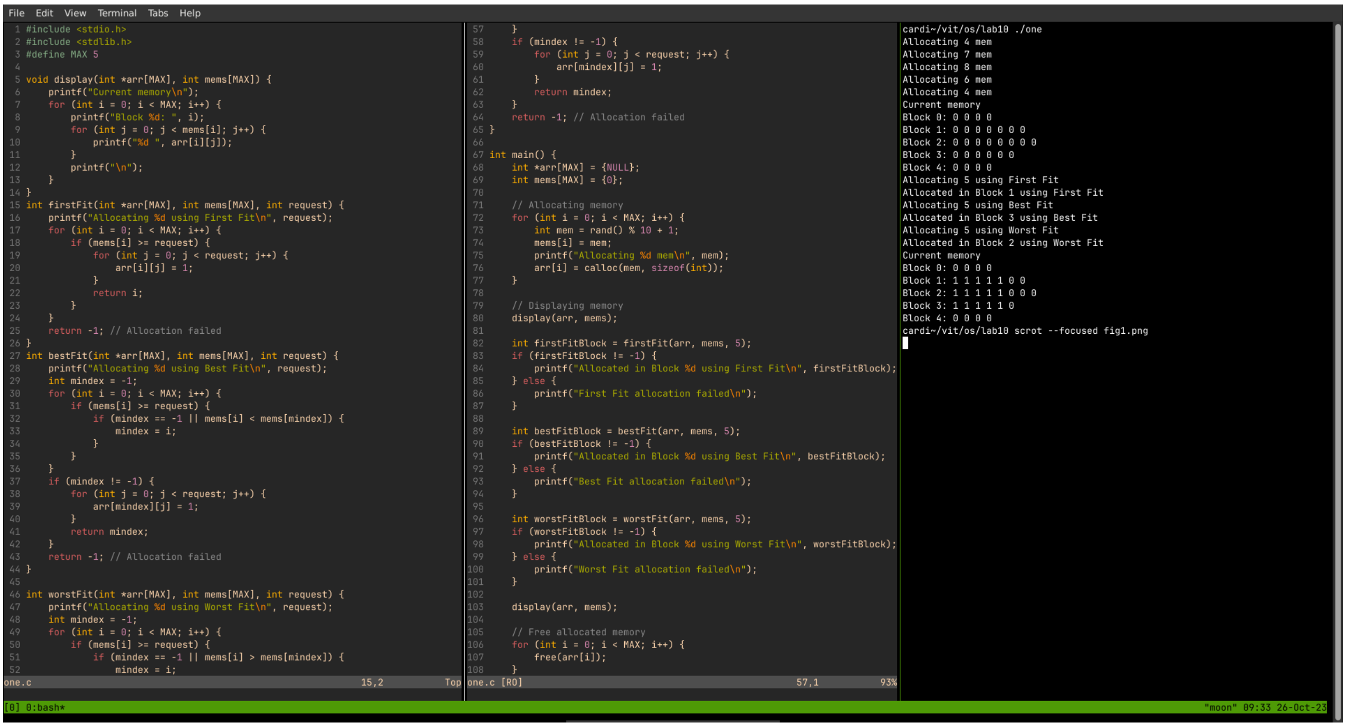
22bps1059

one

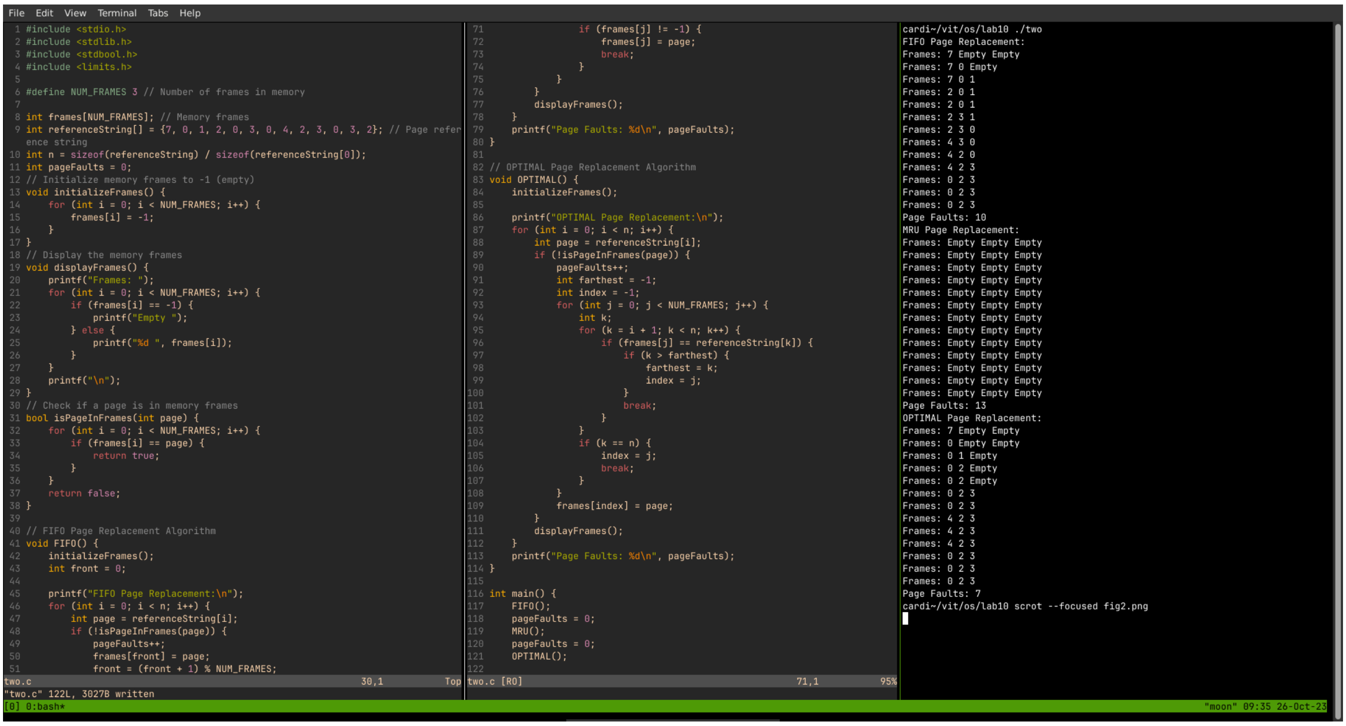


```c
#include <stdio.h>
#include <stdlib.h>
#define MAX 5

void display(int *arr[MAX], int mems[MAX]) {
    printf("Current memory\n");
    for (int i = 0; i < MAX; i++) {
        printf("Block %d: ", i);
        for (int j = 0; j < mems[i]; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
}
int firstFit(int *arr[MAX], int mems[MAX], int request) {
    printf("Allocating %d using First Fit\n", request);
    for (int i = 0; i < MAX; i++) {
        if (mems[i] >= request) {
            for (int j = 0; j < request; j++) {
                arr[i][j] = 1;
            }
            return i;
        }
    }
    return -1; // Allocation failed
}
int bestFit(int *arr[MAX], int mems[MAX], int request) {
    printf("Allocating %d using Best Fit\n", request);
    int mindex = -1;
    for (int i = 0; i < MAX; i++) {
        if (mems[i] >= request) {
            if (mindex == -1 || mems[i] < mems[mindex]) {
                mindex = i;
            }
        }
    }
    if (mindex != -1) {
        for (int j = 0; j < request; j++) {
            arr[mindex][j] = 1;
        }
        return mindex;
    }
    return -1; // Allocation failed
}

int worstFit(int *arr[MAX], int mems[MAX], int request) {
    printf("Allocating %d using Worst Fit\n", request);
    int mindex = -1;
    for (int i = 0; i < MAX; i++) {
        if (mems[i] >= request) {
            if (mindex == -1 || mems[i] > mems[mindex]) {
                mindex = i;
```

```c
    }
    if (mindex != -1) {
        for (int j = 0; j < request; j++) {
            arr[mindex][j] = 1;
        }
        return mindex;
    }
    return -1; // Allocation failed
}

int main() {
    int *arr[MAX] = {NULL};
    int mems[MAX] = {0};

    // Allocating memory
    for (int i = 0; i < MAX; i++) {
        int mem = rand() % 10 + 1;
        mems[i] = mem;
        printf("Allocating %d mem\n", mem);
        arr[i] = calloc(mem, sizeof(int));
    }

    // Displaying memory
    display(arr, mems);

    int firstFitBlock = firstFit(arr, mems, 5);
    if (firstFitBlock != -1) {
        printf("Allocated in Block %d using First Fit\n", firstFitBlock);
    } else {
        printf("First Fit allocation failed\n");
    }

    int bestFitBlock = bestFit(arr, mems, 5);
    if (bestFitBlock != -1) {
        printf("Allocated in Block %d using Best Fit\n", bestFitBlock);
    } else {
        printf("Best Fit allocation failed\n");
    }

    int worstFitBlock = worstFit(arr, mems, 5);
    if (worstFitBlock != -1) {
        printf("Allocated in Block %d using Worst Fit\n", worstFitBlock);
    } else {
        printf("Worst Fit allocation failed\n");
    }

    display(arr, mems);

    // Free allocated memory
    for (int i = 0; i < MAX; i++) {
        free(arr[i]);
    }
```

```
cardi~/vit/os/lab10 ./one
Allocating 4 mem
Allocating 7 mem
Allocating 8 mem
Allocating 6 mem
Allocating 4 mem
Current memory
Block 0: 0 0 0 0
Block 1: 0 0 0 0 0 0 0
Block 2: 0 0 0 0 0 0 0 0
Block 3: 0 0 0 0 0 0
Block 4: 0 0 0 0
Allocating 5 using First Fit
Allocated in Block 1 using First Fit
Allocating 5 using Best Fit
Allocated in Block 3 using Best Fit
Allocating 5 using Worst Fit
Allocated in Block 2 using Worst Fit
Current memory
Block 0: 0 0 0 0
Block 1: 1 1 1 1 1 0 0
Block 2: 1 1 1 1 1 0 0 0
Block 3: 1 1 1 1 1 0
Block 4: 0 0 0 0
cardi~/vit/os/lab10 scrot --focused fig1.png
```

two



```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <limits.h>

#define NUM_FRAMES 3 // Number of frames in memory

int frames[NUM_FRAMES]; // Memory frames
int referenceString[] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2}; // Page reference string
int n = sizeof(referenceString) / sizeof(referenceString[0]);
int pageFaults = 0;
// Initialize memory frames to -1 (empty)
void initializeFrames() {
    for (int i = 0; i < NUM_FRAMES; i++) {
        frames[i] = -1;
    }
}
// Display the memory frames
void displayFrames() {
    printf("Frames: ");
    for (int i = 0; i < NUM_FRAMES; i++) {
        if (frames[i] == -1) {
            printf("Empty ");
        } else {
            printf("%d ", frames[i]);
        }
    }
    printf("\n");
}
// Check if a page is in memory frames
bool isPageInFrames(int page) {
    for (int i = 0; i < NUM_FRAMES; i++) {
        if (frames[i] == page) {
            return true;
        }
    }
    return false;
}

// FIFO Page Replacement Algorithm
void FIFO() {
    initializeFrames();
    int front = 0;

    printf("FIFO Page Replacement:\n");
    for (int i = 0; i < n; i++) {
        int page = referenceString[i];
        if (!isPageInFrames(page)) {
            pageFaults++;
            frames[front] = page;
            front = (front + 1) % NUM_FRAMES;
```

```
two.c                                              30,1            Top
"two.c" 122L, 3027B written
```

```c
                if (frames[j] != -1) {
                    frames[j] = page;
                    break;
                }
            }
        }
        displayFrames();
    }
    printf("Page Faults: %d\n", pageFaults);
}

// OPTIMAL Page Replacement Algorithm
void OPTIMAL() {
    initializeFrames();

    printf("OPTIMAL Page Replacement:\n");
    for (int i = 0; i < n; i++) {
        int page = referenceString[i];
        if (!isPageInFrames(page)) {
            pageFaults++;
            int farthest = -1;
            int index = -1;
            for (int j = 0; j < NUM_FRAMES; j++) {
                int k;
                for (k = i + 1; k < n; k++) {
                    if (frames[j] == referenceString[k]) {
                        if (k > farthest) {
                            farthest = k;
                            index = j;
                        }
                        break;
                    }
                }
                if (k == n) {
                    index = j;
                    break;
                }
            }
            frames[index] = page;
        }
        displayFrames();
    }
    printf("Page Faults: %d\n", pageFaults);
}

int main() {
    FIFO();
    pageFaults = 0;
    MRU();
    pageFaults = 0;
    OPTIMAL();

```

```
two.c [RO]                                         71,1            95%
```

```
cardi~/vit/os/lab10 ./two
FIFO Page Replacement:
Frames: 7 Empty Empty
Frames: 7 0 Empty
Frames: 7 0 1
Frames: 2 0 1
Frames: 2 0 1
Frames: 2 3 1
Frames: 2 3 0
Frames: 4 3 0
Frames: 4 2 0
Frames: 4 2 3
Frames: 0 2 3
Frames: 0 2 3
Frames: 0 2 3
Page Faults: 10
MRU Page Replacement:
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Frames: Empty Empty Empty
Page Faults: 13
OPTIMAL Page Replacement:
Frames: 7 Empty Empty
Frames: 0 Empty Empty
Frames: 0 1 Empty
Frames: 0 2 Empty
Frames: 0 2 Empty
Frames: 0 2 3
Frames: 0 2 3
Frames: 4 2 3
Frames: 4 2 3
Frames: 4 2 3
Frames: 0 2 3
Frames: 0 2 3
Frames: 0 2 3
Page Faults: 7
cardi~/vit/os/lab10 scrot --focused fig2.png
```

```
[0] 0:bash*                                   "moon" 09:35 26-Oct-23
```