Experiment 7
Memory Allocation Methods

Done By: Rohit Karunakaran
Roll No: 58
Date Of Submission: 18-08-2021

Program Code:

```
// Memory allocation methods
// Done By: Rohit Karunakaran
#include<stdio.h>
#include<stdlib.h>
#include<limits.h>

int* getList(int numElem, char fmtStr[]){
    /*Function for reciving numElem integers and returning the array*/
    /*fmtStr: the format string to be displayed while reciving the characters*/
    int *list = (int*) malloc(sizeof(int)*numElem);
    for (int i=0;i<numElem; i++){
        printf(fmtStr,i);
        scanf("%d",list + i);
    }
    return list;
}

void printMemParts(int* list, int numElem){
    /* Displays the memory blocks like this:
     * |---------|--------|
     * | Block 1 | Block2 |
     * |---------|--------|
     * */
    char top[] = "|----------";
    char end[] = "|----------|";

    for(int i=0;i<numElem; i++){
        if(i< numElem-1){
            printf("%s",top);
        }else{
            printf("%s",end);
        }
    }
    printf("\n|");

    for (int i=0; i<numElem; i++){
        printf("%8dKB|",list[i]);
    }
    printf("\n");

    for(int i=0;i<numElem-1; i++){
            printf("%s",top);
    }
    printf("%s",end);
    printf("\n");
}

void show_alloc(int* memList, int mnum, int* procList, int pnum, int*alloc_list){
    /* Displays the processes allocated within each memory block
     * |----------|---------|
```

```
     * |    Block1|  Block 2|
     * |----------|---------|
     * |  P1(45KB)| P0(30KB)|
     * |----------|---------|
     * */
    char start[] = "|--------------";
    char end[]   = "|--------------|";
    char str[30]; // This contains the string Pn(xKB) where n is the process number and x is the size of
the process
    int not_fit[pnum];
    int k=0;
    int total=0;
    int used=0;

    // Print initial |----- and |-----|
    for(int i=0;i<mnum-1; i++){
            printf("%s",start);
    }
    printf("%s",end);
    printf("\n");

    // Print The memory block values
    printf("|");
    for (int i=0;i<mnum; i++){
        sprintf(str,"%dKB",memList[i]); // Integer to string converstion
        printf("%14s|",str);
    }
    printf("\n");

    // Next set of dashes
    for(int i=0;i<mnum-1; i++){
            printf("%s",start);
    }
    printf("%s",end);
    printf("\n|");

    // Print the Process and the size allocated
    for (int i=0; i<mnum; i++){
        if(alloc_list[i] != -1){
            sprintf(str,"%s%d(%dKB)","P",alloc_list[i],procList[alloc_list[i]]); // Get the formatted
string in str
            printf("%14s|",str);
            used += procList[alloc_list[i]];
        }
        else{
            printf("%14s|","NULL");
        }
        total += memList[i];
    }
    printf("\n");

    // Final set of dashes
    for(int i=0;i<mnum-1; i++){
            printf("%s",start);
    }
    printf("%s",end);
    printf("\n");

    // Find process that didn't fit
    for (int i=0; i<pnum; i++){
        int j = 0;
        for (;j<mnum;j++){
            if(alloc_list[j] == i){
                break;
```

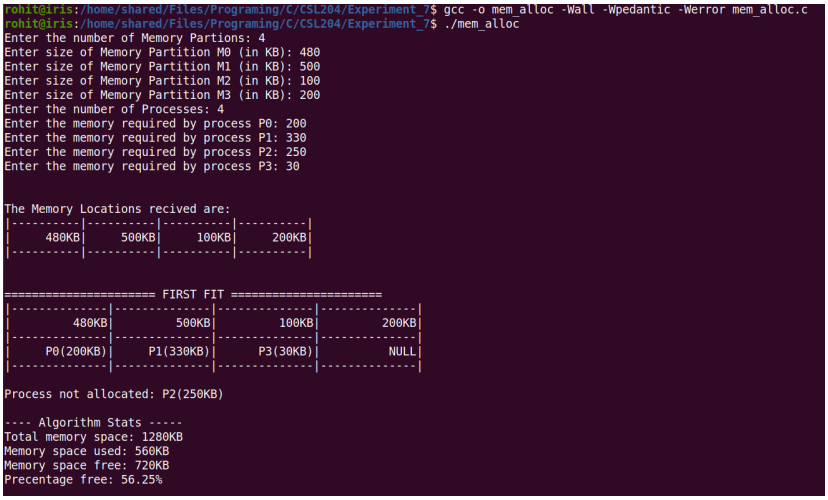
```
            }
        }
        if (j==mnum){
            not_fit[k] = i;
            k++;
        }
    }
    printf("\nProcess not allocated: ");
    for(int i=0;i<k-1;i++){
        printf("P%d(%dKB), ",not_fit[i],procList[not_fit[i]]);
    }
    if (k!=0)
        printf("P%d(%dKB)\n",not_fit[k-1],procList[not_fit[k-1]]); // Print the last unallocated process as
it doesnt require ","
    else
        printf("NIL\n"); // k==0 then all the process fit

    // Some stats (Optional)
    printf("\n---- Algorithm Stats -----");
    printf("\nTotal memory space: %dKB\n",total);
    printf("Memory space used: %dKB\n",used);
    printf("Memory space free: %dKB\n",total-used);
    float precent_free = (float)(total-used) / total * 100.0;
    printf("Precentage free: %.2f%%\n",precent_free);


}

int* first_fit(int* procList, int pnum, int* memList, int mnum){
    int i,j;
    int* alloc_list = (int*)malloc(sizeof(int)*mnum); // will contain the i-1th process in the j-1th
location // -1 if there is no process
    // alloc_list provieds 1-1 mapping with the memList
    // initialise alloc_list to -1 i.e. they are unallocated
    for (i=0;i<mnum;i++){
        alloc_list[i] = -1;
    }

    for (i = 0; i< pnum ; i++){
        //Find a place that fits for the i th process
        for (j =0; j< mnum ; j++){
            if (alloc_list[j] == -1 && memList[j] >= procList[i]){ // Process fits if the seat is big
enough and not taked
                alloc_list[j] = i;
                break;
            }
        }
    }
    return alloc_list;
}

int* best_fit(int* procList,int pnum, int*memList, int mnum){
    int i,j;
    int* alloc_list = (int*)malloc(sizeof(int)*mnum);
    for (i=0;i<mnum;i++){
        alloc_list[i] = -1;
    }

    int space;
    int pos;
    for(i=0;i<pnum;i++){
        space =INT_MAX;
        pos = -1;
        for(j=0;j<mnum; j++){
```

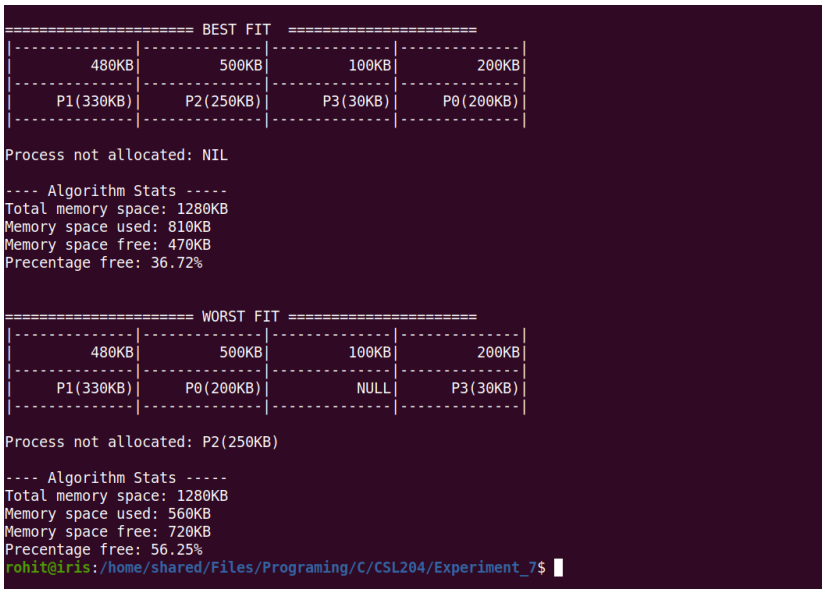
```
            int diff = memList[j] - procList[i];  // The space remaining in the memory block if the process
is allocated
            if(diff≥0 && diff < space && alloc_list[j]==-1){ //The block is elible for allocation if diff
greater than 0 and less than previously calculated diff and the block is not allocated
                space = diff;
                pos = j;
            }
        }
        if (pos ≠ -1){ // if pos is -1 then there is no eligible block
            alloc_list[pos] = i;
        }
    }
    return alloc_list;
}

int* worst_fit(int* procList,int pnum, int*memList, int mnum){
    int i,j;
    int* alloc_list = (int*)malloc(sizeof(int)*mnum);
    for (i=0;i<mnum;i++){
        alloc_list[i] = -1;
    }

    int greatest;
    int pos;
    for(i=0;i<pnum;i++){
        greatest = -1;
        pos = -1;
        for (j=0;j<mnum ; j++){
            if (memList[j]≥procList[i] && greatest < memList[j] && alloc_list[j] == -1){ // find the
greatest unallocated space and return it
                greatest = memList[j];
                pos = j;
            }
        }

        /* No need to calculate diff in worst fit
        for(j=0;j<mnum; j++){
            int diff = memList[j] - procList[i];
            if(diff≥0 && diff > greatest && alloc_list[j]==-1){
                greatest = diff;
                pos = j;
            }
        }*/

        if (pos ≠ -1){
            alloc_list[pos] = i;
        }
    }
    return alloc_list;
}

int main(){
    int n,p;
    printf("Enter the number of Memory Partions: ");
    scanf("%d",&n);
    int *mem_blocks = getList(n,"Enter size of Memory Partition M%d (in KB): ");
    printf("Enter the number of Processes: ");
    scanf("%d",&p);
    int *proc_list  = getList(p,"Enter the memory required by process P%d: ");
    int *alloc_list;

    printf("\n\nThe Memory Locations recived are:\n");
    printMemParts(mem_blocks, n);
```

```
    printf("\n\n====================== FIRST FIT ======================\n");
    alloc_list = first_fit(proc_list,p,mem_blocks,n);
    show_alloc(mem_blocks, n, proc_list, p, alloc_list);
    free(alloc_list);

    printf("\n\n====================== BEST FIT  ======================\n");
    alloc_list = best_fit(proc_list,p,mem_blocks,n);
    show_alloc(mem_blocks, n, proc_list, p, alloc_list);
    free(alloc_list);

    printf("\n\n====================== WORST FIT ======================\n");
    alloc_list = worst_fit(proc_list,p,mem_blocks,n);
    show_alloc(mem_blocks, n, proc_list, p, alloc_list);
    free(alloc_list);

    free(mem_blocks);
    free(proc_list);
    return 0;
}
```

Screenshots:

Sample Input 1





Sample input 2

```
rohit@iris:/home/shared/Files/Programing/C/CSL204/Experiment_7$ ./mem_alloc
Enter the number of Memory Partions: 5
Enter size of Memory Partition M0 (in KB): 50
Enter size of Memory Partition M1 (in KB): 100
Enter size of Memory Partition M2 (in KB): 90
Enter size of Memory Partition M3 (in KB): 200
Enter size of Memory Partition M4 (in KB): 50
Enter the number of Processes: 4
Enter the memory required by process P0: 90
Enter the memory required by process P1: 20
Enter the memory required by process P2: 50
Enter the memory required by process P3: 200


The Memory Locations recived are:
|----------|----------|----------|----------|----------|
|      50KB|     100KB|      90KB|     200KB|      50KB|
|----------|----------|----------|----------|----------|


====================== FIRST FIT ======================
|--------------|--------------|--------------|--------------|--------------|
|          50KB|         100KB|          90KB|         200KB|          50KB|
|--------------|--------------|--------------|--------------|--------------|
|      P1(20KB)|      P0(90KB)|      P2(50KB)|     P3(200KB)|          NULL|
|--------------|--------------|--------------|--------------|--------------|

Process not allocated: NIL

---- Algorithm Stats -----
Total memory space: 490KB
Memory space used: 360KB
Memory space free: 130KB
Precentage free: 26.53%
```

```
====================== BEST FIT  ======================
|--------------|--------------|--------------|--------------|--------------|
|          50KB|         100KB|          90KB|         200KB|          50KB|
|--------------|--------------|--------------|--------------|--------------|
|      P1(20KB)|          NULL|      P0(90KB)|     P3(200KB)|      P2(50KB)|
|--------------|--------------|--------------|--------------|--------------|

Process not allocated: NIL

---- Algorithm Stats -----
Total memory space: 490KB
Memory space used: 360KB
Memory space free: 130KB
Precentage free: 26.53%


====================== WORST FIT ======================
|--------------|--------------|--------------|--------------|--------------|
|          50KB|         100KB|          90KB|         200KB|          50KB|
|--------------|--------------|--------------|--------------|--------------|
|          NULL|      P1(20KB)|      P2(50KB)|      P0(90KB)|          NULL|
|--------------|--------------|--------------|--------------|--------------|

Process not allocated: P3(200KB)

---- Algorithm Stats -----
Total memory space: 490KB
Memory space used: 160KB
Memory space free: 330KB
Precentage free: 67.35%
rohit@iris:/home/shared/Files/Programing/C/CSL204/Experiment_7$
```

Sample Input 3

```
rohit@iris:/home/shared/Files/Programing/C/CSL204/Experiment_7$ ./mem_alloc
Enter the number of Memory Partions: 5
Enter size of Memory Partition M0 (in KB): 50
Enter size of Memory Partition M1 (in KB): 100
Enter size of Memory Partition M2 (in KB): 90
Enter size of Memory Partition M3 (in KB): 200
Enter size of Memory Partition M4 (in KB): 50
Enter the number of Processes: 7
Enter the memory required by process P0: 100
Enter the memory required by process P1: 20
Enter the memory required by process P2: 30
Enter the memory required by process P3: 70
Enter the memory required by process P4: 100
Enter the memory required by process P5: 50
Enter the memory required by process P6: 30


The Memory Locations recived are:
|----------|----------|----------|----------|----------|
|      50KB|     100KB|      90KB|     200KB|      50KB|
|----------|----------|----------|----------|----------|


====================== FIRST FIT ======================
|--------------|--------------|--------------|--------------|--------------|
|          50KB|         100KB|          90KB|         200KB|          50KB|
|--------------|--------------|--------------|--------------|--------------|
|      P1(20KB)|     P0(100KB)|      P2(30KB)|      P3(70KB)|      P5(50KB)|
|--------------|--------------|--------------|--------------|--------------|

Process not allocated: P4(100KB), P6(30KB)

---- Algorithm Stats -----
Total memory space: 490KB
Memory space used: 270KB
Memory space free: 220KB
Precentage free: 44.90%
```

```
====================== BEST FIT  ======================
|--------------|--------------|--------------|--------------|--------------|
|          50KB|         100KB|          90KB|         200KB|          50KB|
|--------------|--------------|--------------|--------------|--------------|
|      P1(20KB)|     P0(100KB)|      P3(70KB)|     P4(100KB)|      P2(30KB)|
|--------------|--------------|--------------|--------------|--------------|

Process not allocated: P5(50KB), P6(30KB)

---- Algorithm Stats -----
Total memory space: 490KB
Memory space used: 320KB
Memory space free: 170KB
Precentage free: 34.69%


====================== WORST FIT ======================
|--------------|--------------|--------------|--------------|--------------|
|          50KB|         100KB|          90KB|         200KB|          50KB|
|--------------|--------------|--------------|--------------|--------------|
|      P5(50KB)|      P1(20KB)|      P2(30KB)|     P0(100KB)|      P6(30KB)|
|--------------|--------------|--------------|--------------|--------------|

Process not allocated: P3(70KB), P4(100KB)

---- Algorithm Stats -----
Total memory space: 490KB
Memory space used: 230KB
Memory space free: 260KB
Precentage free: 53.06%
rohit@iris:/home/shared/Files/Programing/C/CSL204/Experiment_7$
```