Lab Exercise – 12

AIM :: Implement `Fit Algorithms` for memory management on Linux using Shell Scripting.

Theory ::

Fit Algorithms

Fit algorithms are essential techniques in operating systems used for memory management, focusing on how memory blocks are allocated to processes. The main types of fit algorithms include First Fit, Best Fit, and Worst Fit, each with distinct characteristics and trade-offs.

1. First-Fit Algorithm

- **Definition**: Allocates the first available block of memory that is large enough for the process.
- **Speed**: Generally faster as it stops searching once a suitable block is found.
- **Fragmentation**: Can lead to fragmentation, leaving small, unusable memory segments.
- **Complexity**: Low time complexity due to its straightforward scanning approach.
- **Use Case**: Suitable for systems with frequent and varied memory requests.

2. Best-Fit Algorithm

- **Definition**: Searches for the smallest block that can accommodate the process to minimize wasted space.
- **Efficiency**: Reduces fragmentation by utilizing smaller blocks effectively.
- **Search Time**: Can lead to longer search times as it must check all blocks.
- **Overhead**: Increased computational overhead due to multiple comparisons.
- **Memory Management**: May result in small unusable fragments over time.

3. Worst-Fit Algorithm

- **Definition**: Allocates the largest available block to the process to keep larger blocks free for future allocations.
- **Fragmentation**: Aims to reduce fragmentation by preserving larger free blocks.
- **Efficiency**: Can lead to inefficient memory use if smaller processes take larger blocks.
- **Time Complexity**: Higher time complexity due to the need to find the largest block.
- **Use Case**: Useful in specific scenarios where maintaining larger blocks is crucial.

1. First-Fit Algorithm

Source_Code ::

```
echo "Amit Singhal - 11614802722 (5C6)"

# Input the number of memory blocks
echo -n "Enter the number of memory blocks: "
read block_count

# Input the sizes of the memory blocks
echo -n "Enter size of blocks: "
read -a block_size

# Initialize block allocation status
block_allocated=()
for ((i=0; i<block_count; i++)); do
    block_allocated[$i]=0
done

# Input the number of processes
echo -n "Enter the number of processes: "
read process_count

# Input the sizes of the processes
echo -n "Enter size of processes: "
read -a process_size

# Allocate memory using First Fit
process_allocated=()
for ((i=0; i<process_count; i++)); do
    allocated=0
    for ((j=0; j<block_count; j++)); do
        if [ ${block_size[$j]} -ge ${process_size[$i]} ] && [ ${block_allocated[$j]} -eq 0 ]; then
            block_allocated[$j]=1  # Mark block as allocated
            process_allocated[$i]=$((j+1))  # Store allocated block index
            allocated=1
            break  # Exit loop once block is allocated
```

```
        fi
    done
    if [ $allocated -eq 0 ]; then
        process_allocated[$i]="NA"  # Mark as not allocated if no block found
    fi
done

# Display the output in tabular format
echo -e "\n+-------------+--------------+-----------+"
echo -e "| Process No. | Process Size | Block no. |"
echo -e "+-------------+--------------+-----------+"
for ((i=0; i<process_count; i++)); do
    printf "| %-11s | %-12s | %-9s |\n" "$((i+1))" "${process_size[$i]}" "${process_allocated[$i]}"
done
echo -e "+-------------+--------------+-----------+"
```

Output ::

```
singhal-amit@singhal-amit-ThinkPad-T430:~$ vi amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ chmod +x amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ ./amit.sh

Amit Singhal - 11614802722 (5C6)

Enter the number of memory blocks: 5
Enter size of blocks (in kb): 100 500 200 300 600
Enter the number of processes: 4
Enter size of processes(in kb): 212 417 112 426

+-------------+--------------+-----------+
| Process No. | Process Size | Block no. |
+-------------+--------------+-----------+
|      1      |     212      |     2     |
|      2      |     417      |     5     |
|      3      |     112      |     3     |
|      4      |     426      |    NA     |
+-------------+--------------+-----------+
```

2. Best-Fit Algorithm

Source_Code ::

```
echo "Amit Singhal - 11614802722 (5C6)"

# Input the number of memory blocks
echo -n "Enter the number of memory blocks: "
read block_count

# Input the sizes of the memory blocks
echo -n "Enter size of blocks: "
read -a block_size

# Initialize block allocation status
block_allocated=()
for ((i=0; i<block_count; i++)); do
    block_allocated[$i]=0
done

# Input the number of processes
echo -n "Enter the number of processes: "
read process_count

# Input the sizes of the processes
echo -n "Enter size of processes: "
read -a process_size

# Allocate memory using Best Fit
process_allocated=()
for ((i=0; i<process_count; i++)); do
    best_fit_index=-1
    for ((j=0; j<block_count; j++)); do
        if [ ${block_size[$j]} -ge ${process_size[$i]} ] && [ ${block_allocated[$j]} -eq 0 ]; then
            if [ $best_fit_index -eq -1 ] || [ ${block_size[$best_fit_index]} -gt ${block_size[$j]} ]; then
                best_fit_index=$j  # Update best fit index
            fi
        fi
    done
    if [ $best_fit_index -ne -1 ]; then
        block_allocated[$best_fit_index]=1  # Mark block as allocated
        process_allocated[$i]=$((best_fit_index+1))  # Store allocated block index
    else
```

```
        process_allocated[$i]="NA"  # Mark as not allocated
    fi
done

# Display the output in tabular format
echo -e "\n+-------------+--------------+-----------+"
echo -e "| Process No. | Process Size | Block no. |"
echo -e "+-------------+--------------+-----------+"
for ((i=0; i<process_count; i++)); do
    printf "| %-11s | %-12s | %-9s |\n" "$((i+1))" "${process_size[$i]}" "${process_allocated[$i]}"
done
echo -e "+-------------+--------------+-----------+"
```

Output ::

```
singhal-amit@singhal-amit-ThinkPad-T430:~$ vi amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ chmod +x amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ ./amit.sh

Amit Singhal - 11614802722 (5C6)

Enter the number of memory blocks: 5
Enter size of blocks (in kb): 100 500 200 300 600
Enter the number of processes: 4
Enter size of processes(in kb): 212 417 112 426

+-------------+--------------+-----------+
| Process No. | Process Size | Block no. |
+-------------+--------------+-----------+
|      1      |     212      |     4     |
|      2      |     417      |     2     |
|      3      |     112      |     3     |
|      4      |     426      |     5     |
+-------------+--------------+-----------+
```

3. Worst-Fit Algorithm

Source_Code ::

```
echo "Amit Singhal - 11614802722 (5C6)"

# Input the number of memory blocks
echo -n "Enter the number of memory blocks: "
read block_count

# Input the sizes of the memory blocks
echo -n "Enter size of blocks: "
read -a block_size

# Initialize block allocation status
block_allocated=()
for ((i=0; i<block_count; i++)); do
   block_allocated[$i]=0
done

# Input the number of processes
echo -n "Enter the number of processes: "
read process_count

# Input the sizes of the processes
echo -n "Enter size of processes: "
read -a process_size

# Allocate memory using Worst Fit
process_allocated=()
for ((i=0; i<process_count; i++)); do
   worst_fit_index=-1
   for ((j=0; j<block_count; j++)); do
      if [ ${block_size[$j]} -ge ${process_size[$i]} ] && [ ${block_allocated[$j]} -eq 0 ]; then
         if [ $worst_fit_index -eq -1 ] || [ ${block_size[$worst_fit_index]} -lt ${block_size[$j]} ]; then
            worst_fit_index=$j  # Update worst fit index
         fi
      fi
   done
   if [ $worst_fit_index -ne -1 ]; then
      block_allocated[$worst_fit_index]=1  # Mark block as allocated
      process_allocated[$i]=$((worst_fit_index+1))  # Store allocated block index
   else
```

```
        process_allocated[$i]="NA"  # Mark as not allocated
    fi
done

# Display the output in tabular format
echo -e "\n+-------------+--------------+-----------+"
echo -e "| Process No. | Process Size | Block no. |"
echo -e "+-------------+--------------+-----------+"
for ((i=0; i<process_count; i++)); do
    printf "| %-11s | %-12s | %-9s |\n" "$((i+1))" "${process_size[$i]}" "${process_allocated[$i]}"
done
echo -e "+-------------+--------------+-----------+"
```

Output ::

```
singhal-amit@singhal-amit-ThinkPad-T430:~$ vi amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ chmod +x amit.sh
singhal-amit@singhal-amit-ThinkPad-T430:~$ ./amit.sh

Amit Singhal - 11614802722 (5C6)

Enter the number of memory blocks: 5
Enter size of blocks (in kb): 100 500 200 300 600
Enter the number of processes: 4
Enter size of processes(in kb): 212 417 112 426

+-------------+--------------+-----------+
| Process No. | Process Size | Block no. |
+-------------+--------------+-----------+
|      1      |     212      |     5     |
|      2      |     417      |     2     |
|      3      |     112      |     4     |
|      4      |     426      |    NA     |
+-------------+--------------+-----------+
```