Homework 4 OS 3500-002

Q1.

1. Logical address does not physically exist and is created by the CPU and can be accessed by the user, Physical address physically exists in the memory unit that cannot be accessed by user.
2. Contiguous storage allocation is when consecutive blocks of information are assigned to a process, non-continuous storage allocation is when the different parts of a process are allocated different places in the main memory.
3. In first-fit placement is when the process is assigned to the first block large enough to store it, best fit placement is when the process is assigned minimal space, only enough for what it needs. This is far more efficient than first fit.

Q2.

First-fit: 17

Best-fit: 17

Next-fit: 17

Worst-fit: 25

Q3.

1. Overlays allocation storage has less time and memory requirements.
2. Compaction reduces external fragmentation and makes memory more efficient.

Q4.

|  |  |
| --- | --- |
| Page | Free Frames |
| 0 | 6 |
| 1 | 7 |
| 2 | 10 |
| 3 | 12 |
| 4 | 18 |
| 5 | 20 |
| 6 | 21 |

1. Logical Map

|  |  |
| --- | --- |
| 0 | .aa, .bb, .cc, .dd |
| 1 | .ee, .ff, .gg, .hh |
| 2 | .ii, .jj, .kk, .ll |
| 3 | .mm, .nn, .oo, .pp |
| 4 | .qq, .rr, .ss, .tt |
| 5 | .uu, .vv, .ww, .xx |
| 6 | .yy, .zz |

Physical Map

|  |  |
| --- | --- |
| 0 | .aa, .bb, .cc, .dd |
| 1 | .ee, .ff, .gg, .hh |
| 2 | .ii, .jj, .kk, .ll |
| 3 | .mm, .nn, .oo, .pp |
| 4 | .qq, .rr, .ss, .tt |
| 5 | .uu, .vv, .ww, .xx |
| 6 | .yy, .zz |



|  |  |
| --- | --- |
| Page | Free Frames |
| 0 | 6 |
| 1 | 7 |
| 2 | 10 |
| 3 | 12 |
| 4 | 18 |
| 5 | 20 |
| 6 | 21 |

Bb: 25

Ff: 29

Rr: 73

Vv: 81

1. Internal fragmentation is 2 because there are 2 spots unused on the 7th page

Q5.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 8 | 5 | 7 | 6 | 4 | 3 |
| - | - | 6 | 6 | 6 | 6 | 6 | 6 | 3 |
| - | 7 | 7 | 7 | 7 | 7 | 7 | 4 | 4 |
| 8 | 8 | 8 | 8 | 5 | 5 | 5 | 5 | 5 |
| F | F | F | √ | F | √ | √ | F | F |

1. 6 Faults, 6/9 = 2/3 = %66

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 8 | 5 | 7 | 6 | 4 | 3 |
| - | - | 6 | 6 | 6 | 7 | 7 | 7 | 3 |
| - | 7 | 7 | 7 | 5 | 5 | 5 | 4 | 4 |
| 8 | 8 | 8 | 8 | 8 | 8 | 6 | 6 | 6 |
| F | F | F | √ | F | F | F | F | F |

1. 8 Faults, 8/9 = %88.8

Q6.

1. It will not likely improve CPU utilization because the CPU is already underutilized
2. It will not likely improve CPU utilization since there is already enough storage
3. This will decrease CPU utilization since less memory is available
4. This will improve CPU utilization by reducing the number of faults
5. This will improve CPU utilization because there is more memory
6. This will not help because the disk that needs upgrading is the paging disk
7. This will help cu utilization by avoiding faults
8. This will not help CPU utilization by reducing the available frames

Q7.

1. Page fault rate = # page fault occurred / # instructions executed
2. Page fault rate + # page fault occurred / # of pages accessed

Q8.

designers could use a virtual memory with a large logical memory, or a designer could

Implement the global page replacement policy.