Javier Escareno

Professor Li

CSCI 114

April 24, 2023

1.)

        In paged memory architecture internal fragmentation happens when a page frame allocated to a process is larger than the actual amount of memory needed by the process. This results in wasted memory as the remaining space cannot be used by any other process. If page frame sizes are too small, this can lead to an excessive number of page faults. This can result in slow performance as the operating system must retrieve the required page from secondary storage. If the page frame sizes are too large, this can lead to increased internal fragmentation, as well as decreased overall system efficiency due to memory waste.

2.)

        When the operating system reuses memory, it must first zero out the contents of the memory or disk to make sure that sensitive data from the previous process is not available to the new process (ex: passwords).If the new process is able to access this data it could lead to security breaches or data leaks.

3.)

        Some of the benefits of using paging as the lowest level of architecture are its flexibility to allocate physical memory to processes in more flexible units, the protection it provides against any unauthorized access requests, and the way paging is able to efficiently manage memory for the operating system.

4.)

LRU Policy

A	C	B	D	B	A	E	F	B	E	A	G	E	F	A
A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
	C	C	C	C	C	E	E	E	E	E	G	G	G	G
		B	B	B	B	B	B	B	B	B	B	B	E	E
			D	D	D	D	F	F	F	F	F	F	F	F
				H	H	F	F	H	H	H	H	F	F	H	H
				C	D						E	B			

Min Optimel

	A	A	A	A	A	A	A	A	A	A	A	A	A
1	A	A	A	A	A	A	A	A	A	A	A	A	A
2	C	C	C	C	C	E	E	E	E	E	E	E	E
3		B	B	B	B	B	B	B	G	G	G	G	
4		D	D	D	D	F	F	F	F	F	F	F	F

Clock Based

A	C	B	D	B	A	E	F	B

```
   A        C        B        D      B     A     E      F      B
1 [A]     1 [A]   1 [A]    1 A    1 A   1 A   1 E   1 E   1 E
→         1 [C]   1 [C]    1 C    1 C   1 C   0 C   1 F   1 F
           →      1 [B]    1 B    1 B   1 B   0 B  →0 B   1 B
                   →       1 D    1 D   1 D   0 D   0 D   0 D
```

```
   F        A        G      E     F     A
1 [E]     1 [E]   0 [E]   1 E   1 E   1 E
1 [F]     1 [F]   0 [F]  →0 F   1 F   1 F
1 [B]    →0 [B]   1 [G]   1 G   1 G   1 G
0 [D]      [A]    1 [A]   1 A   1 A   1 A
```