Ray Omoregie

7/1/20

CST-221

Memory Management

1. When working within a machine, we see that multiple techniques are used in proving space and imputing space. Of the components, two are well known for being used by both machine and operating system. Physical Address is known for being replicated to the max size of any flexible memory to be accessed. Virtual Address is known for working in the Operating System. By the amount of process, it provides the given virtual address amount of a single space per the given process. These are their independent processes and come in a given form of stacks.

We can see that this can transfer to the stack of physical address bit space. We see that are known for being chunks. To fully move and translate the addresses, we see that the given page table which is currently running is being used.

Here is a pseudo-code in which shows and demonstrates the translation between addresses:

A screenshot of a cell phone

Description automatically generated

1. When working with the given space, we see that each unit of memory has its own space. When the diagram below shows the example of the processes being inputted into the physical memory of the program. We see that each has a max point in which how many procedures are illustrated and given. On the side, we see that each space is corresponding to the actual memory unit and then transferred from there. This shows the physical memory being in-store and then converted with the new spaces. In a result, we see that this is the best method for using and inputting space.

A picture containing text

Description automatically generated

1. We see that the separation between Policy and Mechanism is based upon the given system. As used in our modern-day world, we have technology that allows us to adhere to this standard. In this given example of the program, both process points are given exclusively to their respective positions. We see that they can’t go into a similar index. We see that this given separation is known for helping the program run more smoothly. Because of this, we see we have a respected amount of space to perform the given operations.

We see that the page fault is known for having too much space running and requested from the page. When this occurs, we have errors that occur as well. Once the page space is made, we see that this data can now be added to the given memory.

In comparison to the External Pager Function, this is known for having the mechanism to separate managing content from managing memory in the given program. For this, it helps the policy and mechanism being incorporated from a different angle.

Screen of a cell phone

Description automatically generated

Output:

A screenshot of a cell phone

Description automatically generated

Here is the given result with the 0X007x Value:

A picture containing screen, monitor, photo, table

Description automatically generated  
Output:

A screenshot of a cell phone

Description automatically generated