|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 🔑 **Essential** 🔑 | | | |  | 🏆 **Enhanced** 🏆 | | | |
|  |  |  |  |  |  |  |  |  |
| \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |  | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |

**Score: \_\_\_\_\_**

**23 – Logical Memory**

**Activities**

COMP256 – Computing Abstractions

Dickinson College

Spring 2022

Prof. Grant Braught

**Name:**

**Introduction:**

Today’s class introduced the idea of a process’ logical memory, explaining that it is an abstraction that allows each process to operate as if it has its own memory that may even be larger than the physical memory in the machine. We then saw three examples of how the logical memory behaves when programs (in C) performed operations that required the use of the Data, Stack and Heap segments in the process’ logical memory. These activities will review and highlight the most important ideas from class and give you some additional practice with the concepts involved.

**Logical Memory:**

In today’s class the library was used in the campus metaphor to help illustrate the logical memory abstraction. We simplified the library’s job to delivering a resource when requested. The contents of the library were the hardware (i.e. the computer’s memory), the process was a student making a request for a resource, and the librarian was the operating system. The metaphor illustrated how each student (process) seems to have their own entire library (memory) and that that library is much much larger than its physical space would allow.

🔑 1. This question asks you to use your metaphor to explain how logical memory makes it seem like each process has their own memory and how that memory is much larger than it actually is.

a. In a sentence, reintroduce your metaphor. You can copy this from a previous assignment. I ask for it to help me remember what it was as I read the rest of your answer.

b. Identify the elements of your metaphor that you will be using to play the roles of the hardware (i.e. the memory), the process and the operating system.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **System Element** | **Metaphor Element** |  |
|  | Memory |  |  |
|  | Processes |  |  |
|  | Operating System |  |  |
|  |  |  |  |

c. Using the elements of your metaphor from part b, explain the how the OS can make it seem like each process has its own logical memory.

d. Using the elements of your metaphor from part b, to explain the how the OS can make it seem like the memory is much larger than it actually is.

e. In the library metaphor, a process always received a resource that it requested. However, it took longer to receive the resource if it was not in the physical library. Discuss how your metaphor also exhibits this property.

**Logical Memory Structure:**

🔑 2. The maximum size of a memory (i.e. the size of its address space) is determined by the number of bits used for the memory addresses. For example, in the K&S the memory addresses were 5 bits each. Thus, the K&S had a 5-bit address space meaning it could address 25 = 32 different memory locations. In real machines, memory addresses use many more bits.

a. Early personal computers based on the Intel 8086 processor used 20 bits for its physical memory addresses.

i. How many different physical memory locations could be addressed by the 8086 processor?

ii. If every address held 1 byte of data, then how many total bytes can be addressed by an 8086 processor? Express your answer in Kilobytes and Megabytes.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **Units** | **Size** |  |
|  | Kilobytes |  |  |
|  | Megabytes |  |  |
|  |  |  |  |

b. Modern personal computers and operating systems use up to 54 bits for logical memory addresses. If each logical address holds 1 byte of data how big would the logical address space be if logical addresses are 54 bits? Express your answer in Gigabytes (GB).

c. How many bits would we need to use for logical memory addresses if we want our system to have a logical address space of 1TB?

🔑 3. The Logical memory of a process is made up of four segments.

a. Which two segments are a fixed size and do not change as the process is executing?

b. Which two segments change size as the process executes?

c. What are some examples of the types of things that are stored in the stack segment?

d. What are some examples of the types of things that are stored in the heap segment?

**Memory Allocation:**

🔑 4. In a C program:

a. What does the malloc function do?

b. What does the value returned by malloc indicate?

c. What does the free function do?

🏆 d. What is a dangling reference (a.k.a. dangling pointer) and how does it occur in a C program?

🏆 e. There is another memory allocation function in C named calloc, that is very similar to malloc, what does it do? (Use google).

**C Program Examples:**

5. Consider the C program on Repl.it at: <https://repl.it/@braughtg/MemoryAllocationHW>. For each of the following variables, indicate if its value will be in the Data, Heap or Stack segment.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **Variable** | **Location** |  |
|  | low |  |  |
|  | ac |  |  |
|  | nc |  |  |
|  | c |  |  |
|  | ac->miles |  |  |
|  | ac->plate |  |  |
|  | nc->miles |  |  |
|  | nc->plate |  |  |
|  |  |  |  |

🔑 6. Draw a picture of the logical memory for the process at the point when the program reaches line 18 (just before the return statement in the getCar function executes). **Use a format similar to the one shown in the class slides. Be sure to show all values and references.** Paste in a picture of your drawing for this question.

🔑 7. Draw a picture of the logical memory for the process at the point when the program reaches line 24 (just before the if statement in the printCar function is executed). **Use a format similar to the one shown in the class slides. Be sure to show all values and references.** Paste in a picture of your drawing for this question.

**A Little OS History:**

Watch the end of *Operating Systems* from the Crash Course series with Carrie Anne, that we started earlier. The end of the video touches a variety of OS topics that includes logical memory (she uses the term Virtual Memory – which we’ll see more precisely next class). It will reinforce some of the things you saw in today’s class, while hinting at what we’ll see next class. It will also discuss some of the other important advances in operating systems and a few of the people behind them.

* <https://www.youtube.com/watch?v=26QPDBe-NB8&t=4m47s> (8:49)

🏆 8. What is memory protection?

🏆 9. What operating system was the first to make use of protected and virtual (logical) memory?

🏆 10. What is a Terminal?

🏆 11. What is time-sharing and which OS pioneered this technique?

🏆 12. Who were the two researchers who created the Unix operating system?

🏆 13. What is the Operating system *kernel*?

Optional: To help me improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.