

Exercise problems on System calls used for Memory Management

Problem 1: Memory Allocation

Write a C program that uses the ``malloc()'` function to allocate memory for an integer array of size ``n'` (user input). Then, prompt the user to enter ``n'` integers, store them in the allocated memory, and print the sum of the integers.

Problem 2: Dynamic Array Resizing

Write a C program that uses ``malloc()'` to create a dynamic integer array. Allow the user to add elements to the array one by one. When the array becomes full, use ``realloc()'` to double its size, and continue accepting elements. Implement this until the user decides to stop, and then print the elements of the dynamic array.

Problem 3: Memory Mapping

Write a C program that uses the ``mmap()'` function to create a memory-mapped file. Allow the user to enter text input, which will be written to the memory-mapped file. Then, read the content of the memory-mapped file and print it.

Problem 4: Shared Memory

Write a C program that utilizes shared memory for inter-process communication. Create two processes using ``fork()'`. One process should write a message to a shared memory segment, and the other process should read and print the message from the shared memory segment.

Problem 5: Memory Locking

Write a C program that demonstrates memory locking and unlocking using ``mlock()'` and ``munlock()'`. Allocate a large block of memory, lock it, perform some operations on it, and then unlock it. Measure the time it takes for the operations with and without memory locking.

Problem 6: Memory Deallocation

Write a C program that allocates memory using ``malloc()'` and then deallocates it using ``free()'`. Create an array of integers with a size determined by user input. After processing the data, make sure to free the allocated memory to prevent memory leaks.

Problem 7: Process Address Space

Write a C program that uses the ``sbrk()'` function to manipulate the process's data space. Implement a simple memory allocation and deallocation system using ``sbrk()'`. Allow the user to allocate and free memory blocks, and print the current data space size.