Charlie Misbach

Computer Architecture

Project #1 Document

3/12/24

**Setup and Main Function**

* The program starts by setting up the necessary data in the .data section, including the sizes of the arrays and the arrays themselves.
* The main function begins by loading the size of the arrays and then calls the merge function to merge arr1 and arr2 into output.
* After merging, it prints the contents of output array.
* Next, it calls the merge2 function to merge arr1 and a buffered version of arr2 (bufferedarr) in reverse order, and then it prints bufferedarr.

**Merge Function**

* The merge function compares elements from arr1 and arr2 one by one and copies the smaller (or equal) element to the output array.
* It keeps track of indexes for both input arrays and the output array to ensure elements are copied correctly.
* If one of the arrays is fully processed before the other, it copies the remaining elements from the other array to the output.
* The comparison is done by the compare function, which determines if one element is less than, equal to, or greater than the other.

**Reverse Merge Function (merge2)**

* The merge2 function performs a similar merge operation but in reverse, starting from the end of the arrays and working backwards.
* It's used to merge the original arr1 with bufferedarr, which already contains arr2 elements, effectively reversing the order of merged elements back into bufferedarr.
* This function also relies on the compare function to decide which element to copy next.

**Compare Function**

* The compare function is a simple utility that takes two numbers and determines their relative sizes.
* It returns -1 if the first number is less than the second, 1 if the first is greater, and 0 if they are equal.

**Printing and Exiting**

* The program includes loops to print the results of the merges to the console.
* It uses system calls to print integers and characters (like spaces and newline characters) to format the output.
* Finally, the program exits cleanly by making an exit system call.