Lab 9

Memory: Row-Major And Column-Major Array Storage

Purpose

To understand storage of arrays in memory and the difference between row-major order and column-major order.

Background Reading And Preparation

Read Chapters 9 through 11 to learn about basic memory organization and the difference between storing arrays in row-major order and column-major order.

Overview

Instead of using built-in language facilities to declare two-dimensional arrays, implement two C functions, *two_d_store* and *two_d_fetch*, that use linear storage to implement a two-dimensional array. Function *two_d_fetch* takes six arguments: the base address in memory of a region to be used as a two dimensional array, the size (in bytes) of a single entry in the array, two array dimensions, and two index values. For example, instead of the two lines:

```
int  d[10,20];
x = d[4,0];
```

a programmer can code:

```
char  d[200*sizeof(int)];
x = two_d_fetch(d, sizeof(int), 10, 20, 4, 0);
```

Function *two_d_store* has seven arguments. The first six correspond to the six arguments of *two_d_fetch*, and the seventh is a value to be stored. For example, instead of:

```
int  d[10,20];
d[4,0] = 576;
```

a programmer can code:

```
char  d[200*sizeof(int)];
two_d_store(d, sizeof(int), 10, 20, 4, 0, 576);
```

Procedure And Details (checkmark as each is completed)

1. Implement function *two_d_store*.
2. Create an area of memory large enough to hold an array, initialize the entire area to zero, and then call *two_d_store* to store specific values in various locations. Use the hex dump program created in Lab 6 to display the result, and verify that the correct values have been stored.
3. Implement function *two_d_fetch*.
4. Verify that your implementation of *two_d_fetch* works correctly.
5. Test *two_d_store* and *two_d_fetch* for boundary conditions, such as the minimum and maximum array dimensions.

Optional Extensions (checkmark as each is completed)

6. Verify that functions *two_d_store* and *two_d_fetch* work correctly for an array that stores: characters, integers, or double-precision items.
7. Extend *two_d_store* and *two_d_fetch* to work correctly with any range of array index. For example, allow the first index to range from -5 to +15, and allow the second index to range from 30 to 40.