CS 310 Terminal Driver

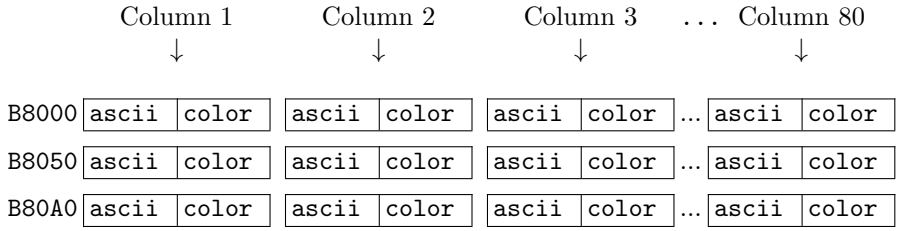
Spring 2026

1 Getting Started

In this homework assignment, you're going to build a terminal driver for your kernel that allows it to print to the terminal. This homework will be done in your OS repo, not as a program in Linux.

The i386 PC has a special memory region reserved for terminal output. To write characters to the screen, we just need to write ASCII characters into video memory.

Video memory is basically just a large array that starts at address `0xB8000`. Each element of the array is two bytes long: one byte tells the computer what ASCII character to display, and the second byte tells the computer what color to use. The first element of the array controls the character that is displayed in the upper left corner of the screen—the first character on the top row. The second element of the array (at address `0xB8002`) controls the character in the second position on the first row, and so on. Each row is 80 characters wide. A diagram of video memory is shown below.



2 Deliverables

1. Write a function called `putc()` that writes a single character to the terminal:

   void putc(int data);

   Successive calls to `putc()` should write characters to the terminal at sequential locations on the screen. It shoul not keep writing characters to the upper left corner of the screen. To make this work, you need to keep track of the offset of the last character that you wrote to video memory and increment that offset each time you write a new character. The best way to do this is to maintain two global variables `int x` and `int y` that keep track of the next position where a character will be written on the screen. When you write a character to screen, increment `int x`. When you reach the end of the line (`x >= 80`), increment `y` and set `x` back to 0.

2. Write a function `puts()` that writes a NULL-terminated string to the terminal by calling `putc()` to print each character in the string.

3. When you reach the bottom of the screen (past the 24th row of characters), your `putc()` function should scroll the contents of the screen up. Do this by overwriting the first row of characters of the screen with the second row, overwrite teh second row with the contents of the third, and so on.