

CZ1106 Problem Solving and Computation II

Programming Lab 2

Turbo C Debugger and Bit Interleaving

1 Feb 2007

Part-I

1. The program ac2-4.c, used to pack the day, month and year into a word, may be downloaded from web.

- Use debugger to add the following watches:
day, month, year, packed, word, n, mask.
- Execute the program step by step by the F7 key. Enter input as **19/11/1994**

Answer the following questions while stepping on the instructions:

- i. What are the values of day before and after day <=&11 is executed ?
Draw a bit map to explain the results.
 - ii. What are the values of month before and after month <=&7 is executed ?
Draw a bit map to explain the results.
 - iii. What are the values of year before and after year%100 is executed ?
- Set break point at word <=&1; Add watch for i. Execute 16 iterations.
- iv. For each iteration of the function print_bit, what are the values of word before and after word <=&1 is executed ?
 - v. Explain why word becomes 0 when i=15.
- Execute the remaining instructions step by step.
- vi. What is the value of packed ?
 - vii. What is the value of day after packed >> 11 is executed ?
Draw a bit map to explain the result.
 - viii. What is the value of month after
month=(packed&0x780)>>7 is executed ?
Draw a bit map to explain the result.
 - ix. What is the value of year after year=packed&0x007f is executed ?
Draw a bit map to explain the result.
2. Before you compile the program ac2-6.c, select word alignment by these steps:
Option, Compiler, Code generation, Word alignment, OK

- Run the program and observe the screen output.
- Now, deselect word alignment. Run the program and observe the screen output again. Explain the cause of the differences in the contents of outputs.

Part-II

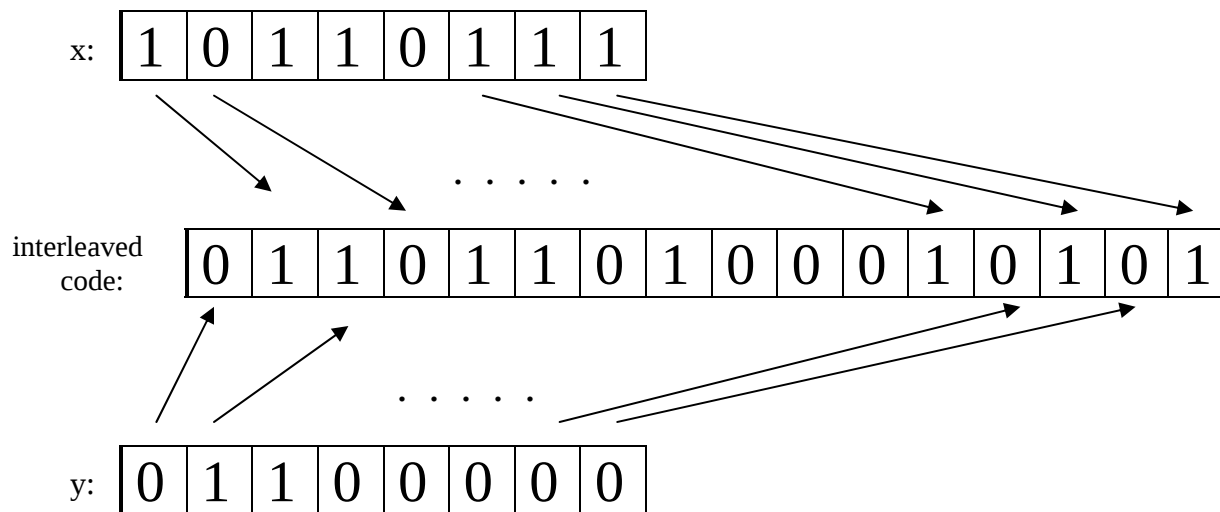
Bit-wise interleaving is often used in spatial database to map a multi-dimensional coordinate to a 1-dimensional coordinate. For example, the 2-dimensional coordinate (183, 96) is mapped to 27925 as illustrated below.

(All base-2 notations are unsigned.)

$$x = 183 = 10110111_{(2)}$$

$$y = 96 = 01100000_{(2)}$$

$$\text{interleaved code} = 0110110100010101_{(2)} = 27925$$



Write a program named as **merge.c** to read two integers from the keyboard and display the decimal and the binary contents of their interleaved contents. The program will repeat the execution until the user selects not to continue.

Assume that the x, y values are not greater than 255. Consequently, your program will have to repeat the data entry if x or y is greater than 255. Once the coordinate is within the assumed range, you can cast the values to unsigned characters. The interleaved contents obtained from the unsigned characters should be stored in an unsigned integer.

The bit interleaving can be performed as follows:

Loop 7 times

```
{  
    temp = LSB of x;  
    if temp is equal to 1, set the MSB of interleaved code;  
    push the bit contents of interleaved code to the right by 1 place;  
  
    temp = LSB of y;  
    if temp is equal to 1, set the MSB of interleaved code;  
    push the bit contents of interleaved to the right by 1 place;  
  
    push the bit contents of x to the right by 1 place;  
    push the bit contents of y to the right by 1 place;  
}
```

*if x is equal to 1, set the MSB of interleaved code;
push the bit contents of interleaved code to the right by 1 place;
if y is equal to 1 set the MSB of interleaved code;*

A session of program execution is shown below:

Enter x:2

Enter y:5

```
x = 2  00000010  
y = 5  00000101  
interleaved code = 38  00000000000100110
```

Continue (y/n)?y

Enter x:183

Enter y:96

```
x = 183  10110111  
y = 96  01100000  
interleaved code = 27925  0110110100010101
```

Continue (y/n)?Y

Enter x:300

Enter x:256

Enter x:255

Enter y:256

Enter y:255

```
x = 255  11111111  
y = 255  11111111  
interleaved code = 65535  1111111111111111
```

Continue (y/n)?n

Use debugger !!!