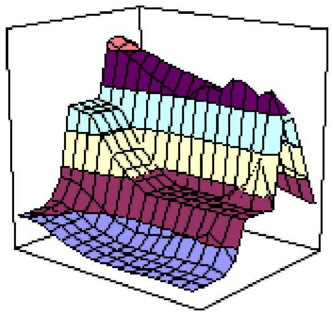

A Note on Byte Operands in ASM

Randal E. Bryant and David R. O'Hallaron

CS:APP Home Page

On page 228 of CS:APP we stated that with GCC's inline assembly "...there are no direct ways to specify a program value to use as the destination operand for the `setae` instruction, since the operand must be a single byte." In fact, you can specify single-byte operands with `gcc` by declaring variables of type `char`. This allows us to simplify the inline assembly for `ok_smul3` (p. 227) to the following (called `ok_smul4`):

```
int ok_smul4(int x, int y, int *dest)
{
    unsigned char byte_result;

    *dest = x*y;

    /* Insert the following assembly code:
       setae byte_result       # Set result
    */
    asm("setae %0"
        : "=r" (byte_result)  /* Output    */
        );

    return (int) byte_result;
}
```

Similarly, here's a simplified version of `ok_umul` (called `ok_umul2`):

```
int ok_umul2(unsigned x, unsigned y, unsigned *dest)
{
    unsigned char byte_result;

    /* Insert the following assembly code:
       movl  x,%eax         # Get x
       mull  y              # Unsigned multiply by y
       movl  %eax, *dest    # Store low-order 4 bytes at dest
       setae byte_result    # Set result
    */
    asm("movl %2,%%eax; mull %3; movl %%eax,%0; setae %1"
        : "=r" (*dest), "=r" (byte_result) /* Outputs    */
        : "r"  (x),     "r" (y)            /* Inputs     */
        : "%eax"                           /* Overwrites */
        );

    return (int) byte_result;
}
```

Thanks to Michael Trigoboff for showing us this trick.

Randy Bryant and Dave O'Hallaron
Last modified: Wed Aug 30 17:49:21 EDT 2006