

# 26

“Riches gotten by doing wrong have no value.”

## Playing with Pointers

Programmers so often praise C for its *pointers*. Pointers are more powerful! In this chapter, let’s see some of the interesting programs that use pointers.

### 26.1 Rebooting with pointers

Believe it or not, using pointers, we can even reboot our system! The following program reveals this.

```
#define BOOT_ADR      (0xFFFF0000UL)
#define RESET_ADR    (0x00400072UL)
#define COLD_BOOT    (0)
#define WARM_BOOT    (1)

void ReBoot( int type ) /* arg 0 = cold boot, 1 = warm */
{
    void ((far *fp)()) = (void (far *)()) BOOT_ADR;
    if ( type==COLD_BOOT )
        *(unsigned int far *) RESET_ADR = 0;
    else
        *(unsigned int far *) RESET_ADR = 0x1234;
    (*fp)( );
} /*--ReBoot( )-----*/

int main( void )
{
    int opt;
    printf( "    Rebooting Program \n\n"
           "Warning: Reboot would result in data loss \a\n"
           "0. Cold Boot \n"
           "1. Warm Boot \n"
           "2. Exit without booting \n"
           "Enter your option: "
           );
    scanf( "%d", &opt );
    if ( opt==0 || opt==1 )
        ReBoot( opt );
    return(0);
} /*--main( )-----*/
```

## 26.2 Identifying machine model and BIOS date

The following program is by **Bill Buckels**. It finds the model of our PC and BIOS date using pointers!

```

/* getmodel.c by bill buckels 1990 */

/* This Program will Provide The Model Of The PC */
/* and its BIOS Release Date by peeking around at */
/* The Top Of The BIOS. */

#undef MK_FP
#undef peekb

#include <stdlib.h> /* required for malloc */
#include <stdio.h> /* required for printf */

/* undefine the above if they exist */
/* all compilers start on equal footing */
/* macros to peek into memory */
/* dynamically cast a far pointer from segment and offset info */
#define MK_FP(seg,off) ((char far *)(((long)(seg) << 16) | (off)))

/* return a byte from a dynamically cast location in memory */
#define peekb(a,b) (*(char far*)MK_FP((a),(b)))

/* memory address information */
#define ROMSEG 0xf000
#define ID_OFFSET 0xffff
#define MD_OFFSET 0xffff5

/* an array of characters */
char idbytes[10]={
    '\x00', '\x9A', '\xFF', '\xFE', '\xFD',
    '\xFC', '\xFB', '\xFA', '\xF9', '\xF8'};

/* an array of strings */
char *idstrings[]={
    "Not In Our List",
    "a COMPAQ plus",
    "an IBM PC",
    "a PC XT or Portable PC",
    "a PC jr.",
    "a Personal Computer AT or PS/2 Model 50 or 60",
    "a PC XT after 1/10/86",
    "a PS/2 Model 30",

```

## 120 A to Z of C

```
        "a Convertible PC",
        "a PS/2 Model 80",
        NULL};

/* a record structure to organize our data */
/* this new data object is called a MODELINFO */

typedef struct{
    unsigned char modelbyte;
    char idinfo[66];
}MODELINFO;

char *captions[3]={
    "\nGETMODEL.EXE by Bill Buckels 1990\n\n",
    "This Computer is ",
    "The BIOS release date is "};

void getmodelinfo(void)
{
    /* a pointer to our MODELINFO's info */
    MODELINFO *modelinfo;

    int num_records = 10 ; /* number of records in the data base */
    unsigned char byte ; /* counters */
    unsigned char mdl,num ;
    char datestring[9] ; /* string space for the date */
    char datelimit=8 ;

    /* allocate the memory in the near heap */
    modelinfo = malloc(num_records*sizeof(MODELINFO));

    /* and fill the memory with the data in our arrays */
    /* an example for use of indirection in structures */

    for(byte=0;byte<num_records;byte++)
    {
        modelinfo[byte].modelbyte = idbytes[byte];

        strcpy(modelinfo[byte].idinfo,
            idstrings[byte]);
    }

    /* get the ID byte */
    num = peekb(ROMSEG, ID_OFFSET);
    mdl = 0;
```

```
/* point to the matching entry in the structure */
for(byte=0;byte<num_records;byte++)
    if(num==modelinfo[byte].modelbyte)mdl=byte;

/* now get the date of the bios */
/* and add it to our date string */
for(byte=0;byte<datelimit;byte++)
    datestring[byte]=peekb(ROMSEG,MD_OFFSET+byte);

/* terminate the string with a null character */
datestring[datelimit]='\x00';

/* print the model info, then the BIOS date */
printf("%s%s\n",
        captions[1],
        modelinfo[mdl].idinfo);
printf("%s%s\n",
        captions[2],
        datestring);
/* and now we are done */
}

int main( void )
{
    puts(captions[0]);
    getmodelinfo();
    return(0);
}
```