

**Part VIII**  
**Next Step**

### **What do you think of C#?**

I have no comments on C# as a language. It will take a lot to persuade me that the world needs yet another proprietary language (YAPL). It will be especially hard to persuade me that it needs a language that is geared for a specific proprietary operating system...

**—Bjarne Stroustrup**, Creator of C++

Courtesy: Bjarne Stroustrup's FAQ

# 68

"Rulers should not desire beer."

## 32-bit Compiler

Today, 32-bit applications and Operating Systems have replaced the existing 16-bit applications and Operating Systems. So people refer the 16-bit environment as obsolete!

### 68.1 16-bit Compiler

16-bit compiler uses 16-bit instruction set to produce 16-bit applications. As we know, 16-bit applications work in real mode. TC++3.0 is a 16-bit environment and it works in real mode. So some people refer TC++3.0 as older C compiler!

### 68.2 32-bit Compiler (DJGPP)

Introduction of 32-bit processor necessitates the need for a 32-bit protected mode C compiler. Thereafter many 32-bit C compilers were introduced. MSDOS port of the GNU C/C++ compiler named DJGPP (by D.J. Delorie and few others) is the winner among other 32-bit compilers. DJGPP provides Unix style of writing C/C++ programs under MSDOS. The free DJGPP compiler and other supporting tools are available under GNU's General Public License, and so source codes are also available!!!

Quite honestly, nowadays most of the DOS programmers use DJGPP than TC++3.0 for developing DOS programs. DJGPP is available on CD-ROM! Please checkout the CD-ROM for installation of DJGPP and documentation. I don't want to go into the details of DOS programming with DJGPP, and it is left to the reader as an exercise! Reader must be aware that 16-bit version of DJGPP was also introduced by D.J. Delorie, but it is not at all used.

Following are the important advantages of DJGPP:


1. DJGPP is a non-proprietary environment for developing 32-bit protected mode software in C/C++ under MS-DOS.
2. DJGPP is good for writing DOS utilities.
3. Very good for Graphics / Game Programming

Personally, I think it is wise to switch to 32-bit compiler than to stick onto 16-bit compiler (TC++3.0). It is left to you to take decision on compilers! If you still want to work with 16-bit compilers, I personally recommend TC++3.0.

#### 68.2.1 Allegro

Allegro is a library of functions for use in computer games, written in a mixture of C and assembly language especially for DJGPP compiler. Allegro is also free as DJGPP. It is found on

## 628 A to Z of C

CD . Allegro provides so many functions to access graphics cards and sound cards. So Allegro reduces the programming effort by enormous amount. Following are the important features of Allegro as by their documentations:

1. Supports VGA **mode 13h**, mode-X (twenty three tweaked VGA resolutions plus unchained 640x400 Xtended mode), and **SVGA modes** with 8, 15, 16, 24, and 32 bit color depths, taking full advantage of VBE 2.0 linear framebuffers and the VBE/AF hardware accelerator API if they are available.
2. Drawing functions including putpixel, getpixel, lines, rectangles, flat shaded, gouraud shaded, and texture mapped polygons, circles, floodfill, bezier splines, patterned fills, masked, run length encoded, and **compiled sprites**, blitting, bitmap scaling and rotation, translucency/lighting, and text output with proportional fonts. Supports clipping, and can draw directly to the screen or to memory bitmaps of any size.
3. Hardware scrolling, mode-X split screens, and palette manipulation.
4. **FLI/FLC animation player**.
5. Plays background **MIDI** music and up to 64 simultaneous sound effects, and can record sample waveforms and MIDI input. Samples can be looped (forwards, backwards, or bidirectionally), and the volume, pan, pitch, etc, can be adjusted while they are playing. The MIDI player responds to note on, note off, main volume, pan, pitch bend, and program change messages, using the General MIDI patch set and drum mappings. Currently supports Adlib, SB, SB Pro, SB16, AWE32, MPU-401, ESS AudioDrive, Ensoniq Soundscape, and software wavetable MIDI.
6. Easy access to the **mouse**, keyboard, **joystick**, and high resolution timer interrupts, including a vertical retrace interrupt simulator.
7. Routines for reading and writing LZSS compressed files.
8. Multi-object data files and a grabber utility.
9. **Math functions** including fixed point arithmetic, lookup table trig, and 3d vector/matrix manipulation.
10. **GUI** dialog manager and file selector.