Table of Contents for The Zen of Assembler

Part III: The Flexible Mind

Chapter 16: Knowledge Matters Only when It's Used

The programmer's integration of knowledge and application is the key to good software...two levels: 1) making the most effective code locally (local optimization); 2) matching that to the application (global optimization)...no sharp line between the two...key is always to know what the PC can do, then match that to the task as efficiently as possible, even when that means using unorthodox techniques...we'll look at an example, then review a number of general and specific principles for "zenning" code (define "zenning")...zen in big ways (program structure, algorithms) & little ways (clever test & jump)

Chapter 17: Executing Zen: A Case History

"Zenning" the simple example from Turbo Pascal Solutions

Chapter 18: Limit Scope as Needed to Match Available Resources

Use buffers <=64K in size, to allow speedy searching and manipulation, paging in data with restartable string instructions if necessary to support this...reduce resolution or color selection if there's not enough memory otherwise...in short, look for ways to pare the program back to the essentials if that's what it takes to run well on a PC...example of redirected file filtering versus internally buffered filtering versus block string filtering

Chapter 19: Be Willing to Break Your Own Rules When Necessary

Don't always preserve all registers...don't stick to parameter-passing conventions when it's not worth it

outside loops...parameter passing...be clever with high/low bit testing (rotate, shift, sign test)...boolean logic & binary arithmetic...and bx,xxxx to both convert to word and mask off