Programming Tools

by Chris Gehlker

So, you want to be a hacker? Maybe you promised your husband (wife, main squeeze) that the Mac would pay for itself; or you've been through the dungeon a couple of times, balanced your checkbook, DLed every picture of members of the opposite sex that you could find, and you still want to find something to do with your Mac. At this point, most hard-core Macers take up programming or start a desktop publishing business out of their homes. I don't know much about the DP business except that yes you can make some money if you have a laser printer and some skill at selling. The programming business is still pretty open too but you have to get a little of a reputation before someone will actually pay you. Fortunately a reputation as a programmer is pretty easy to get. Just write a program and give it away. Make sure you put your name, address and phone number in the about box. It doesn't hurt to say that you are looking for work.

One upon a time, if you had a Fat Mac and an assembler you were in Fat City as far as programming tools went. Except for the full-time nerds at Microsoft and Apple with their Lisas, that was it for programming tools. Much good code was written that way too. Now the mechanics of programming are a lot easier. There are all kinds of programming tools and aids and some of them are really useful. Hock your car.

Compilers

While there are now several good development systems available for the Mac, Apple's MPW environment, with integrated C, Pascal and Assembler, and Symantec's THINK Pascal and THINK C dominate the market. These are both good systems. Apple's is much more expensive both in terms of the initial purchase price and in terms of the hardware platform needed to run it effectively. The minimum practical required system for MPW is an 020/030 machine with 4 megs and a hard disk. Both THINK C and THINK Pascal will run acceptably on a two-meg Plus, debugger and all. The THINK compilers are fast and produce reasonably good code. The MPW environment is a UNIX clone and is therefore powerful but a bit hard to learn. Get the THINK languages first and the MPW when you can afford it. In fact, get all the real languages that you can afford. Hackers collect languages the way mechanics collect tools. Sooner or later you'll run across a job where having the right language will really simplify your life. How do you recognize a real language? That's easy. Real languages aren't called Basic and they aren't called HyperCard. Using either of these is shameful.

Assembler

MPW is the assembler of choice. If you can't afford it, get MacAssembly which is shareware but a solid product nonetheless. There is also an inline assembler in THINK C which wouldn't be anything to write home about except it's inline. You will have to learn to read Assembler and the best way to do that is to write some; so do get an assembler. Don't kid yourself though; if you want to get paid to program, at some point or other you will probably have to write some Assembler. That's what you'll think at first. Later you'll realize that "get to write assembler" is the appropriate phrase. Assembler is fun.

Resedit

This wonderful program comes with most development systems. Use it: Master it: Customize it: Make it your own. Get the docs! If you can't find them on a BBS, pay APDA for them. Many development systems also come with RMaker. Throw it away. If you want a text based resource editor, pay the bucks for MPW and get Rez or use the stand alone version that comes with THINK Pascal.

Debuggers

A while ago the choice was easy. There was TMON which you used if you were serious and MacsBug which you used if you were cheap. Then a funny thing happened; MacsBug got a whole lot better. As I write this MacsBug is up to version 6.1 and it supports multiple monitors and even color. Its best feature is its ability to support macros which you can make on the fly or build into it with ResEdit. Couple this with the ability to create templates for any data structure that you can devise make it a real winner.

A good high level debugger is also necessary. Think C and Pascal have, very different, debuggers built in. If you use the MPW package, be sure to get SADE.

Books

You will have to get all five volumes, soon to be six, of Inside Macintosh. This is the Bible for Mac programming. I still think the Macintosh Revealed series by Chernicoff is the best introduction to coding for the Mac. You won't learn Pascal from this book; it assumes that you already know it. Macintosh Programming Primer is supposedly very good on both C and the Mac toolbox. Dan Weston's books on assembler programming are also very good. If you don't know C or Pascal, by all means get a good tutorial on your chosen language and work through all the examples before you tackle Mac programming.

Other Neat Stuff

By all means get the Programmer's Online Companion from Addison Wesley. It pops up a little window at the bottom of your screen that shows the functional form for all the Mac toolbox calls and all the more important structures known to the Mac ROM. There are well over 6,000 Toolbox calls and you won't remember how to spell all of them, let alone the types and order of all their arguments.

Get Started

Once you have all of the stuff above, you are ready to earn some money programming. There are many other neat things that you could get and I'll talk about them later but the toolkit that I've laid out above is really plenty. Write a "Hello World!" program and then write the great freeware game or utility. Once you've done that, work won't be hard to find.