talking about buying a building, when the checkbook hasn't been balanced in 6 months! Just amazing. . . .

I replaced my Osborne keyboard recently with a homebrew version made from a surplus keyboard. My typing speed has gone up to about 70 wpm!

Thanks for the nail files--is this a subtle hint? Hey, maybe I could grow them like--Fu Manchu!!! Naw, Mike's beat me to it. . . .

As far as the computer business is concerned, I think waiting is still the name of the game. I'm really most interested in developing software--this area has a large pool of underemployed talent that can be tapped, and the expenses involved in retailing machines are just too high. The expense of purchasing a machine alone is daunting (at least $7500, all at once), and I'm inclined to wait a bit to see how the near future develops. AT&T, for example, will be releasing its very first commercial computer system "someday soon," and the reactions of other firms may be interesting (like lowering prices!). The ideal solution is to get access to somebody else's machine until prices do come down a little--of course, that basically means getting a job, which isn't exactly easy around here.

However, a lot of new machines are becoming available which are attractive for development work (i.e. very powerful); this could be a factor in attracting people to work with me as co-developers (hey, I certainly won't be able to pay them anything!). I think this is important; I'm hoping to get some good input from the people I've contacted at Cornell on attractive directions, and hope that I can enlist people with skills complementary to mine. Unfortunately, those machines are even more expensive (around $10,000 minimum).

It really is a crazy business--the uncertainty is really incredible to contemplate. It's not just that you need to produce a relatively unique and reliable product that meets somebody's needs; after that, you've got to market it, in a market with a million vendors howling out the virtues of their product(s) to consumers who are not only confused but also have a bad habit of stealing your software. It may have cost you a lot to develop it, but it costs almost nothing to copy it; other vendors also steal a lot.

Gee, I hope I'm not making you too nervous. . . .

One angle I'm considering quite seriously is writing about computers and "programming"; every time somebody asks about, say, an introductory text on programming, I'm stuck; I just don't know of anything I'd recommend which is both introductory-level and useful. Most texts get bogged down in details immediately, and never offer an overview which tells people why things are done a certain way. One of the unusual things about UNIX is that it assumes that the people using the system are intelligent enough to solve their own problems if they're given the right tools--learning to **use** the tools together, though, involves a fair amount of education. Therefore, the market for UNIX educational materials is going to be pretty good, assuming that UNIX does go over the way it should; and since I'm better at writing and thinking in general than at programming specifically, this field may offer a better application of my skills. (One of the super-nifty things about UNIX is that text entered under the system can be typeset