The Mac: Personality vs. Performance

by Dave Kosiur

I like running INITs and cdevs -- this capability to personalize your Mac is one of the curses of this machine however. Adding an INIT here or a cdev there desn't seem to affect the system much but if you start accumulating a list of over 20 INITs/cdevs, as I have, things start to appear sluggish.

Naturally, I could disable all my INITs, but I like having double-headed scroll arrows and 3-D colored icons! So, in an attempt to determine whether performance or personality was more important, I embarked on a little research into my INITs.

Once before, I had used a public domain program called CheckTicks to find out which INITs were "stealing" time from my normal actions on the Mac. (CheckTicks counts the number of ticks it takes to run through a loop of 10,000 GetNextEvent calls.) By and large, I found that many INITs had little effect on the system, but those responsible for intercepting key strokes or other related events were usually the ones taking the largest time requirements, which stands to reason. So I dusted off CheckTicks again and used to investigate my current system configuration (which is a Mac II with System 6.0.7 running with MultiFinder and a 128K cache). Here's a summary of what I found --

(1) The base system (as above, with the DeskTop Manager and GateKeeper INITs on) gave me a starting value of 521 ticks. Many other INITs, such as DeskPict!, FileSaver (Norton Utilities), and SunDesk, gave me times within a few ticks of the base value, so, by themselves, they weren't a major problem.

(2) CE ToolBox was a noticeable culprit, requiring a run time of 770 ticks. (After all, it has to sit there, testing for command-key combinations.)

(3) SuperBoomerang, which requires CE ToolBox, also was a "laggard" at 838 ticks.

(4) Perhaps the worse offender was ClickChange. Running it in addition to my base configuration resulted in CheckTicks requiring 1028 ticks to complete. However, most of ClickChange's options are cosmetic, i.e., cursor shapes, scroll design, etc., which shouldn't affect the tick count. The one option that I thought was slowing things down was probably the sound option, since ClickChange had to filter certain types of events to decide if a sound should be played. Turning off the sound option in ClickChange resulted in CheckTicks only requiring 547 ticks to complete -- a noticeable improvement!

(5) Remember that the execution of all these INITs is cumulative. Take the case where one INIT adds 200 ticks to CheckTicks' execution while a second INIT adds 50 ticks to CheckTicks' execution. If you use both INITs together, they will add 250 ticks to CheckTick's execution. When you've got a long list of INITs running (as I do), even those adding a few ticks to the execution loop start adding up.

(6) But I like sounds on my Mac -- at least for a few things (Startup and Shutdown, mainly). So I dusted off SoundMaster (which I'd removed when I bought ClickChange) and re-installed it to provide the same sounds that I'd used with ClickChange. Checking SoundMaster with my configuration resulted in a CheckTicks execution time of 538 ticks. Certainly that was livable. To check the

configuration I was most interested in, I tried SuperBoomerang + ClickChange (with the sound option on) versus SuperBoomerang + ClickChange (with sound off) + SoundMaster. In the first case, the tick count was 1428 ticks! In the second case, the tick count was 886. Guess which configuration I picked?