MAC FACTS

from

Mac Help Desk

SUPPORT, SALES, TRAINING & SERVICE

(972) 783-9787 • (972) 783-7550 - *Fax/Modem* • (214) 249-9543 - *Pager*
e-mail address - machelpdesk@home.com
Web site - www.machelpdesk.com

Volume 8, Number 5 May 1999

A Message from Dru

As in years past, May is always busy here at Rancho Richman. Did you call your Mom on Mother's Day? I did! On the 7th, step-daughter Hillary returned for a few days from Univ. of Nebraska where she just (successfully) completed her fresh(wo)man year. On the 9th, other step-daughter Alisha was one of 5 juniors attending the Richardson High School Senior Prom and on the 10th she successfully earned her Brown Belt in Tai Kwan Do (so watch out or she'll kick your butt). On the 12th, my lovely child-bride celebrated her birthday [with as much pomp as I can muster]. And lastly Carol and I will celebrate our third wedding anniversary on the 26th and then go to Branson, Mo. to party.

❍❁❁❁❁❁▼▲ ❁⏢❍❁❊●

I guess it's time to remind everyone (again!) that we are NOT a 24/7 [24 hours a day,7 days a week] company. Our office hours are 8:30 am to 5:30 pm Monday through Friday and 8:30 am to 1:30 pm on Saturday. After those hours I remove my pager and turn the ringers off on the phones and spend some time with my family. We are closed on Sundays.

❍❁❁❁❁❁▼▲ ❁⏢❍❁❊●

If April showers bring may flowers, what do may flowers bring? Why pilgrims, of course. Here are April's pilgrims - Peter Blackburn, Al Creemer, D Media, Bill Preston, Linda Hartley, Sandy Ammons, Part Hurt, Marilyn Cowley, Active Concepts, Odeneal & Odeneal, Al Thom, Mike Gleason, Sara Gates, Kathy Steele, Janice Gardner, Otis Bunger, Lilian Prather, Roger Christensen, Barbara Tyler, Carrington Labs, the Dallas Zoo, Susan Krasnow, Karlene Justice, Beverly Gibbons, Thomas Svitek, Natural EFx, and Rev. Donald Parish. Welcome all.

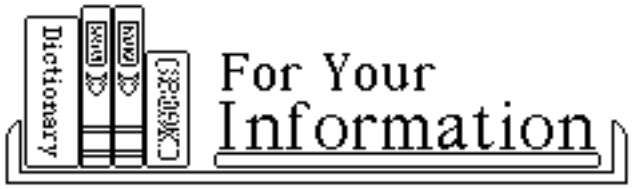
❍❁❁❁❁❁▼▲ ❁⏢❍❁❊●

Don't forget - the Apple Corps of Dallas' monthly meeting at Big Town Exhibition Hall in Mesquite on Saturday May 15th. Starts at 8:30. Meet other Mac users, discuss all kinds of stuff, and don't forget the vendors. This month's presenter is Adobe. They'll be speaking about Adobe's fantastic new program, InDepth. Also don't forget my radio show, CyberLine, is on every Sunday night on WBAP (820 on your AM dial) from 7-9 pm. Tune in, listen, call in. On May 23rd we will have the Mac Doctor himself, Bob LeVitas, as a guest. We can play stump the guest. :-)

❍❁❁❁❁❁▼▲ ❁⏢❍❁❊●

Know someone who'd like to receive MacFacts? Get me their name and e-mail (or fax) number and I'll take care of it

❍❁❁❁❁❁▼▲ ❁⏢❍❁❊●



Power Macintosh G3: The Cannonball Express

by Rick Holzgrafe

The Cannonball Express was the fabled train that was so fast it took three men to say "Here she comes," "Here she is," and "There she goes." Computers are fast too, although unlike trains, most aren't self-propelled. What makes a computer fast, and how much effect does software design have? How much faster are today's computers than yesterday's? Recently I revisited some of these questions, beginning with a trip down memory lane.

Back in the Stone Age -- Twenty years ago, I was teaching myself programming and had access to a DEC PDP-11/60 minicomputer on evenings and weekends. This beast was bigger than a washing machine, and during workdays I shared it with two dozen other technicians and engineers. I found a word puzzle in a magazine and thought it would be fun to program the PDP to solve it. The puzzle was as follows.

Given a phrase and a sheet of graph paper, write the phrase on the graph paper according to these rules:

1. Write one letter per square on the graph paper, like filling in a crossword puzzle. Ignore case and anything that's not one of the 26 letters of the alphabet. The phrase "N. 1st Street" is thus identical to "NSTSTREET".

2. Put the first letter of the phrase in any square you like.

3. After writing any letter, put the next letter of the phrase in any adjacent square. Here, "adjacent" means any of the eight neighboring squares, up, down, left, right, or diagonally. You may reuse a square if it is adjacent and already holds the letter you need; otherwise you must use a blank square. You can't use the same square twice in a row - no "hopping in place" for double letters.

The goal is to write the phrase inside a rectangle of the smallest possible area. (A subtle point: you are not trying to write in a minimal number of squares.) To score your solution, draw the smallest enclosing rectangle you can and take its area. The rectangle may enclose some blank squares; count them, too.

Got it? Tongue-twisters are the most fun because they have lots of opportunities to reuse whole snaky strings of squares. The 37 letters in "Peter Piper picked a peck of pickled peppers" can be packed into a 3 by 5 rectangle, like this (view this in a monospaced font):

```
OFIPT
KCPER
LEDAS
```

In those days I knew computers were "fast" but had no idea how fast. The answer turned out to be "not very." I wrote a program to solve these puzzles and called it Piper after the tongue-twister. I set Piper running on a medium-length phrase on a Friday evening, and came back on Monday to find it still running. It had found several less-than-best solutions but hadn't finished. Way too slow - I found a better solution myself on paper in about half an hour.

Why did it take so long? Piper was a "brute force" program. It tried every possible solution to the problem, one after another. The trouble is that there are too many possible solutions. Exactly how many depends on the phrase, but for any non-trivial phrase the number is astronomical. I realized for the first time that "fast" sometimes isn't "fast enough." This point may be obvious today, when we all use computers and are weary of waiting for them. But in 1979, that PDP-11 was only the second computer I had ever seen!

What Part of Fast Don't You Understand? I saw that I would have to make Piper faster. There are two basic ways to speed up a program. Plan A is to find a better way of solving the problem, but after twenty years I still haven't thought of a better solution. That leaves plan B, the classic efficiency expert's solution: eliminate unnecessary steps. For example, Piper created every possible solution, then calculated the area of each. It built each solution one letter at a time, so instead of taking the area only for completed solutions, I changed Piper to check the area after placing each letter. If placing a letter made the solution-in-progress take up more space than the smallest complete solution found so far, Piper could skip the rest of that solution (and all other solutions that started the same way) and move right on to the next one. This eliminated a huge amount of work and greatly improved Piper's speed. Finding clever ways to track the area of a growing solution helped too, because it was faster than calculating the area from scratch after each letter. I also found a way to calculate a minimum size for the final solution quickly: I couldn't guarantee that the best solution would be that small, but I could guarantee that it wouldn't be smaller. If Piper got lucky and found a solution as small as that calculated minimum, it could stop immediately. Otherwise it would continue on after finding the best solution, vainly seeking a still better one.

Eventually Piper became clever enough to finish that original phrase in a reasonably short period of time. But the holy grail continued to elude me: I wanted a solution for "How much wood would a woodchuck chuck if a woodchuck could chuck wood?" That PDP (and, perhaps, my cleverness) were not up to the task. I had run out of ideas for speeding up Piper, and runs still took longer than a weekend. But if I couldn't improve Piper, I could at least hope to run it on a faster computer.

Big Iron -- People tend to think of processor speed as the speed of a computer, but many factors affect overall performance. Virtual memory lets you work on bigger data sets or on more problems at a time, but it's slow, so adding more physical RAM helps by reducing your reliance on virtual memory. Faster disks and I/O buses load and save data more quickly. RAM disks and disk caching replace slow disk operations with lightning-quick RAM access. Instruction and data caches in special super-fast RAM offer big improvements for some programs. Well-written operating systems and toolboxes can outrun poorly written ones.

But in the end, little of this matters to Piper. Piper has always used only a small amount of data, doesn't read or write the disk after it gets going, and does little I/O of any kind. With its small code and data set Piper can take good advantage of data and instruction caching, but what it mostly needs is "faster hamsters" - a faster processor to make the wheels turn more quickly.

As the years rolled on, I ran versions of Piper on my first Macs, but in the middle 1980's I worked for Apple Computer, and had access to a programmer's dream: Apple's $15 million Cray Y-MP supercomputer, one of only two dozen in the world and arguably the fastest computer in existence at the time. I figured the Cray would make short work of Piper. But the Cray was not well suited to the problem. It could barrel through parallel-processing floating-point matrix calculations like the Cannonball Express, but Piper was a highly linear, non-mathematical problem. Piper used only one of the Cray's four processors and didn't do the kind of operations at which the Cray excelled. Piper wasn't a fair test of the Cray's power, but the Cray was still the fastest machine I'd ever used. The Cray succeeded where all previous machines (that PDP, my Mac Plus, my Mac II) had failed. It solved "woodchuck" in less than a day, taking only about 20 hours to finish its run. I was awestruck - 20 hours?! I'd no idea that "woodchuck" was that big a problem!

Young Whippersnappers -- I set Piper aside for many years, but recently I began to wonder how a modern desktop box compares to those old minicomputers and mainframes. I rewrote Piper from memory and ran it on my new 400 MHz ice-blue Power Macintosh G3 with "woodchuck." The output is below. Piper first reprints the phrase, then prints solutions and elapsed times as it finds them. Each solution is the best found so far, culminating in the best of all. The times are in seconds from the beginning of the run; the final time is the total run time. (Unfortunately, the best solution for "woodchuck" is larger than Piper's calculated minimum, so Piper continued to run for a bit after finding the best solution.)

Here are the results. Some intermediate solutions have been left out for brevity, but you can see Piper finding ever smaller solutions. In the end, the 57 letters in "woodchuck" are packed into a 4 by 4 rectangle. Have a look at Piper's total run time, and the time needed to find its best solution:

How much wood would a woodchuck chuck if a woodchuck could chuck wood?

0 seconds:

```
ULD  ADLU
HDOAIUCOHWUHOD
UCOWFKHDOMCWOW
K   WO
```

1 seconds:

```
DLIFADLU
UCKOHWUHOD
OHDWOMCWOW
```

2 seconds:

```
ULCHC
HWUHODKUK
OMCWOWAFI
```

7 seconds:

```
HWUHOW
OMCWOD
IKAUCW
FLDHK
```

9 seconds:

```
HWUH
OMCW
LUOK
HDOI
CWAF
```

65 seconds:

```
HWM
OUC
IKH
FWC
ADO
LUO
```

67 seconds:

```
HWMU
OOCH
UDWK
LAFI
```

Total run time: 107 seconds

There you have it: a shade over a minute to find the best solution, under two minutes to finish its run. Two minutes! So much for the big iron of the 1980's. My new G3 Mac finished "woodchuck" over 600 times faster (and 5,000 times cheaper) than that 15 megabuck Cray

The Future -- What's yet to come? 400 MHz already looks a little pokey. It's the best Apple offers today, but I've seen claims of 550 MHz or so from third party accelerators and over-clocking tricks. People are predicting 1 GHz (1,000 MHz) chips for the near future. Buses are getting faster, and caches hold more data in less space and are moving onto the processor chip for still more speed. (Small is fast. Did you know that the speed of light is a serious limiting factor in modern computer design? The closer together the components are, the faster they can signal each other.)

And like the old Cray, multi-processor desktop systems are starting to appear. They gang up on a problem by having separate processors work on different parts of the problem simultaneously. Although I didn't try to use the Cray's extra processors, I've done a little thinking lately. Piper doesn't have to be completely linear. On an eight-processor system, I bet I could come close to making Piper run in one-eighth of the time of a single processor.

Are you ready?

Here she comes -
Here she is -
There she *GOES!*

❍❁❁❁❁❁▼▲ ❁⏢❍❁❊●

NEWSLINE

Jobs unveils Mac notebooks, OS upgrade

By Jack McCarthy

SAN JOSE -- Buoyed by the recent upturn in Apple Computer Inc.'s fortunes, Steve Jobs, cofounder and interim CEO, yesterday introduced two new notebooks, an upgrade to the Macintosh operating system and an agreement with Sears, Roebuck and Co. to sell the iMac starting Memorial Day weekend.

Speaking at Apple's World Wide Developer Conference, Jobs pointed to high attendance figures at the gathering as a sign of the company's health. More than 2,500 people attended the conference, up 43% from last year's event. It was a record for paid attendance, he added.

"The developers have come to the Mac, and I think that's great," Jobs said. "We are also getting a ton of first-time computer buyers now."

Two new PowerBook notebooks, with 333-MHz and 400-MHz processors, offer significant improvements from previous models, Jobs said. The notebooks weigh 5.9 lbs. They are 2 pounds lighter and almost 20% thinner than their predecessors.

With lithium-ion batteries, the new books run up to five hours on a single battery and up to 10 hours on two batteries, an industry record, Jobs said. The notebooks, Jobs said, are "the fastest product in the marketplace running at desktop speeds."

Some developers said they were impressed by the announcements.

"It's encouraging to see the progress they've made," said Mike Mihalik, vice president of engineering at LaCie Inc. in Hillsboro, Ore., a hardware developer. "They seem to be delivering on their promises."

"Jobs has given us a lot to work with," said Clay Hutcherson, technology director of Centric Systems Corp. in London, Ontario, which uses Macintosh systems. Hutcherson praised the OS 8.6 upgrade in particular.

A PowerBook with a 333-MHz processor, 512K bytes of L2 cache and 56K bit/sec. modem sells for $2,499. A PowerBook with a 400-MHz processor, 1M byte of L2 cache and 56K bit/sec. modem sells for $3,499.

Jobs also introduced the Mac OS 8.6, an updated version of the Macintosh operating system touted as easier to use and with a more powerful version of Sherlock, its Internet search feature. The operating system also has an improved Universal Serial Bus integration.

Mac OS 8.5 customers can download OS 8.6 free at www.apple.com/macos/ or buy it on CD-ROM for $19.95. Shipping will begin immediately.

"We ask ourselves if we should delay this stuff and wait for the release of the Mac OS X, and in the end we thought we couldn't because this stuff is so good," Jobs said.

A beta version of the Mac OS X was introduced yesterday, while the finished version is slated for release in January, Jobs said.

Strong interest in the iMac is powering the popularity of Apple, said Rob Enderle, an analyst at the Santa Clara consulting firm Giga Information Group Inc. "Things are a lot more upbeat than they were a few years ago," Enderle said.