SEPTEMBER 199Ø

VERSION Ø1.Ø3 $2.00

THE OSK'ER®

News and Views in the World of OS9/68000 and 6809

ATLANTA COCOFEST

O SAY CAN YOU C?

and some FICTION...

.login

```
           Directory of /dd/OSKer/Sep90 20:39:22
 Owner  Last modified Attributes Sector Bytecount Name
------- ------------- ---------- ------ --------- ----
Delmar  90/10/17 0009  ----r-wr     1      6763 Another_Machine
Editor  90/10/17 0024  ----r-wr     2     10558 Atlanta_Fest
Editor  90/10/13 1421  ----r-wr     4      4207 BBS_List
Wayne   90/10/19 0122  ----r-wr     5     10529 Fiction
Editor  90/10/17 0025  ----r-wr     7      7087 Hacker_Contest
Fleagle 90/10/19 0123  ----r-wr    10     10585 O_Say_Can_You_C
Editor  90/10/22 1247  ----r-wr    13      9400 Playing_Chess_in_C
Editor  90/10/16 0102  ----r-wr    15     12858 Ramblings
```

The Cover Art for this issue, drawn by Alan Sheltra, depicts the 'Wizard of OSK' in honor of the Hacker Contest conclusion (or start of?).

CONSIDERATIONS FOR SUBMITTORS

There are a few things ya'll can do to make my life a heck of a lot easier. When sending submissions of any type (text, program) to the OSKer, make sure of a few things first:

1) The file(s) are in ASCII text, preferably without linefeeds.
2) Try to keep within 80 columns, but NO JUSTIFICATION!!
3) Do NOT indent paragraphs, but do leave a blank line between them.
4) Please don't leave any left margins or hyphenate any words.
   (i.e. leave the format as you see it printed within here)
5) Be sure to include your name and how to contact you.
6) Best way to send it is via a network: I am sysop@root on StG-net, StG@hummer.iupui.edu on I-net, and 72427,335 on CIS
7) If sending by mail, use OS9 CoCo Format, 5.25" or 3.5" (720k), or ST format 3.5" (720k), or MM1 format 3.5" (1.44M)
8) If you want to archive it, please use the OS9 .AR format.

the OSKer

'the OSKer' is printed monthly by StG Computers inc, P.O. Box 24285, Speedway, Indiana, 46224. The president and editor is Scott Griepentrog, V.P. Jim Hutchins, Secretary Chris Swinefurth, and Treasurer Dave Henk.

Subscriptions to the OSKer are $12 within the U.S., $15 for Canada, and $20 overseas.

Editing and Layout for the OSKer was done solely on a prototype MM1 machine under OSK. uMacs is used for pre-editing of text, and a custom routine is used for formatting and layout. An ALPS Allegro 500 (flat paper path, 24 pin) printer is used to produce the initial copy for duplication by offset, as well as printing the mailing addresses on the back cover. The subscription list is also kept in a database under OSK.

To prevent a conflict of interest, StG Computer inc., as both publisher of the OSKer and having ownership of software, will not directly advertise in this magazine, nor will the editor in any way promote said software.

Another_Machine

To be fair to all, the specs to the latest entry into the OSK market are being published here. Ed Gresick sent me a workup of his new machine just in time for the issue, and spoke a few words of wisdom as well. We discussed the possibility of running Kevin's Windows on his system, and he voice an opinion that he holds very strongly: Any software that REQUIRES windows is not fit to be published. Now there are the obvious exceptions such as drafting programs and the like that simply cannot be done without hi-res graphics, but what he is saying is that programs can use the fancy capabilities of the windowing environment such as mice, hi-res graphics, fonts, and so forth, but they should also be able to run from just a plain dumb terminal. For example, a spreadsheet program can make use of the TERMCAP entry to do a full-screen display, and use the mouse only if it is available. Actually, it just makes sense - anybody writing a program to run under OSK wants to appeal to the most users. If said program will work (albeit not as nice) with just the basic capabilities of the average dumb terminal, more people will buy the software as they can have more people using their machine. Which is the big feature of OSK anyways, right? Having the windows is important, but making the software work even without it whenever possible is also a major consideration. Anyways, on with the fun...

DELMAR CO is pleased to offer the SYSTEM IV 68000 Computer System. Operating at a clock speed of 16 MHz, the SYSTEM IV is a speed whiz - and when you add OS9/68000 capability, low-cost expansion, high density drives, VGA and graphics at a cost-competitive price - the system can't be beat. The SYSTEM IV is a mature product and delivery is made from stock. The reception received at the Atlanta CoCoFest was outstanding!

The SYSTEM IV is delivered to the customer ready for use - only an appropriate monitor is required. Among its outstanding features is expandability using readily available, PC/XT plug-in cards. This allows the user maximum versatility.

The SYSTEM IV is housed in a 'baby AT case', comes with one 3 1/2" drive and is designed to hold three additional 5 1/4" drives. Space is available for an additional 3 1/2" internal drive. All connections to the machine are on the back. These include power, keyboard, parallel printer, 4 serial ports, and monitor. The 'RESET' button is on the front panel. A heavy-duty 200 watt switching power supply is included.

The microprocessor is a Motorola 68000 running at 16 MHz. The board will hold up to 4 Megabytes of DRAM (0 wait state). The I/O chips, the clock chip, and all necessary support chips are on the board. All integrated circuit chips are socketed.

The board has seven PC/XT compatible slots. Two slots are used by the system; one for the hard disk controller and the second for the video board (VGA or Hercules). The remaining five slots are available for the user to add PC/XT cards, available from Computer Stores and mail order houses, to fullfill special needs (such as additional I/O, midi, sound, D/A, mouse driver, etc). As plug-in board improvements occur or requirements change, all that is needed is to unplug the old board and plug in the new. Only the drivers may have to be changed. (Depending upon demand, certain boards will be available from and supported by DELMAR CO in the future.)

The SYSTEM IV was designed and is manufacturered by Peripheral Technology. It is based on their four-layer, K4 board; the latest of their 68000 based products. Other boards designed and produced by Peripheral Technology include their K2 board (an earlier version for the 68000), a 68008 board, a 6809 board, and boards for the 80286. Since its founding in 1978, Peripheral Technology has sold many thousands of the above boards.

The graphics capability of the SYSTEM IV is independent of the 68000 microprocessor. Therefore, the performance of the SYSTEM IV is substantially superior to machines using the 68070/66470 chip set. Kevin Pease, designer of the MM/1, measured the performance of the SYSTEM IV at the Atlanta CoCoFest at 1666 drystones. (Drystones is one of the industry tests used to measure the performance of a processor.) This performance is constant and is independent of the video board installed. This compares with about 1000 drystones for machines using the 68070/66470 chip set.

Either of two video options are provided - Hercules Monochrome or VGA. The Hercules Monochrome option allows use of a standard TTL monochrome monitor. The VGA option allows use of either a VGA color monitor or a VGA monochrome monitor. VGA modes currently supported include: 40 character text, 80 character text, 320 x 200 x 256 graphics, 640 x 200 x 16 graphics, 640 x 350 x 16 graphics, 640 x 480 x 16 graphics, and 800 x 600 x 16 graphics.

Professional OS9/68000 is provided with the SYSTEM IV. This includes the C Compiler, Macro Assembler and Linker, Tape Backup, two Editors and a full set of utilities. File managers, drivers and device descriptors for the devices and I/O ports supplied are installed on the system.

All modules necessary to the operating system are loaded into memory when the system is booted. Since Microware is continually improving the OS9/68000 system, this permits updating as improvements are released (a new ROM does not have to be obtained or burned). Also, the user may easily add and/or customize various managers, drivers and descriptors.

At power-on or reset, a menu is presented permitting the user to select various options; i.e., boot from floppy drive, hard drive or select the drive for auto-boot. Operating systems other than OS9/68000 may be installed at this time; however, these are not supported by DELMAR CO.

The built-in real-time clock has battery backup (over 5 years of battery life) and provides 2 KBytes of non-volatile memory. An AT style, 101 keyboard is provided with the system. The parallel printer port provided on the SYSTEM IV is compatible with most popular printers (standard Centronics). A second 8-bit port and two programmable interrupt timers are provided. Four RS-232 Serial Ports using hardware handshaking are provided. Baud rates are selectable from 300 baud to 38,400 baud. Connectors are DB-25. An additional on-board connector is provided to allow adding a memory board capable of holding up to 8 MBytes of DRAM. This will provide a total of 12 MBytes of memory, which can be especially useful for a a large ram disk(s) or adding a large disk cache system.

SYSTEM IV has a one year parts and labor warranty, is in production and is available for immediate delivery. See our add in this issue for more information.

Atlanta_Fest

You sure this was the first one?

by Scott Griepentrog

I spent 10 hours straight driving from Speedway to Atlanta to get there, but the first annual CoCoFest held there was definately worth it. From having spoken with people before hand, I came to the conclusion early on that this wasn't going to be such a big deal. After all, it was not orgainzed by my illustrious competitor, it was their first one, and it ran only for two days on a weekend bordered by either school or work for the majority of people who might attend. Certainly they couldn't convice enough people to come, both exhibitors and fest'ers, to make it a big enough deal to be worth the effort to get there. And I had a big family get together on the same weekend, so I choose to see some relatives I wouldn't get a chance to again for some time.

But when that trip fell through, I (and for some strange reason, quite eagerly) packed up my provisions, including a computer of course, and took off for Atlanta. The Holiday Inn at which the fest was being held was so easy to find (being all of ten yards from the exit ramp) I missed it the first time. I checked in (at 6 in the morning) and they gave me this card with a magnetic strip on it to get in my room. Hey, this is kool. I was freakin out. Okay, look, it was 6 am, okay? I just drove ten gruesomely boring hours to get there. And I find they actually had these neat digital locks...

Actually, Atlanta itself is quite a boom town of recent. Going across the north side of town I had passed a large number of big name corporate office buildings. To pass some time (I got there a day early) I went downtown and poked through the science museum they have called "SciTrek". It is more suited towards teach basic science to children, but they were about to open a new section (funded by IBM of course) covering some math concepts they didn't teach me in gradeschool. But they did have one very interesting exhibit that caught my eye. As some of you may know, Atlanta will be hosting the '96 Olympics. But what you probably don't know is that they convinced the site selection commitee with a Multi-Media application that is very impressive.

This fancy exhibit consisted of two display screens, stereo speakers, and a trackball. A map of various places around town where certain competitions are to be held is displayed on the lower screen, with a cursor that can be moved via the trackball. No big deal, right? The top screen displayed a moving birds-eye view of the site as you approach it on the map. Move right into the building and it shows you an example of the kind of competition to be held there, complete with sound. In places where they didn't have a facility built there yet, a computer generated image is super-imposed, and then you are taken inside. The graphics were so well done you could see the wood grain of the boards making up the velodrome as they took you for a spin around it. The best part was the way the view flys through the air (obviously done with a helicopter) as you go from site to site. At one point you fly right through a window in a skyscraper, go through some offices, sit down on a couch and watch curtains part to reveal a screen on which is projected an image that turns out to be you flying out the other end of the building. Wild!

Anyways, back to the fest. The booths were literally overflowing from the exhibition room. The very first one had Steve Bjork sitting behind it selling his latest game written specially for the fest. "Marty's Nightmare" is a PacMan-ish game for CoCo3/RSDOS that lets you live his worst fear - having all the info but not enough time to share it. With a screen full of CoCo's, disks, chips, and mice, it daunted fest-goers by announcing "I've got a hardware problem". Want a copy? Sorry - it was sold only at the show (to make it a collectible). More on Steve later.

Frank Hogg had several of his K-Bus machines to show, but obviously absent was either Tomcat board. Manufacturing delays due to changes were cited. More on this later. Next around the loop was Delmar with a new OSK machine featuring a PC bus and using PC cards. See file "Another_Machine".

And of course there was IMS with the MMI. I actually _saw_ a two board machine complete with hard drive running! And Kevin Darling came down and showed a kool animation (actually digitization) of the Enterprise taking off - with the theme music playing at the same time. The whole place was crowded, but people understandably tended to congregrate there at times...

I didn't take enough notes on everybody else, but also exhibiting were Alpha Software (showing their BBS program and other OS9 goodies), Burke & Burke, Carl England, CoCo Corner (offering Lyra), CoCoPro (the sponsor), ColorSystems, DanoSoft, Granite Computer Systems, KB Enterprises (multi-tasking RSDOS!), Microcom, Robert Offerman, Second City (Mike Knudsen showed off new UltiMuse!), Specialty Products, SpectroSystems, Sub-Etha, SuperSoft, T&D Subscription, Three C's Projects, and Zebra Systems. If you weren't there, you missed out!

Especially because of the great seminars. The first I attended was Steve Bjork giving a history of the CoCo and revealing a few little known facts about it's development. He also made the point that CoCo software tends to cost a fraction of that on the PC - an important point not only for people thinking about hold funeral cermonies for their CoCo, but also for OSK developers.

Next up was Cray Augsburg from the Rainbow (Hi guys!) talking about all the negative rumors only hurting everybody. He said he "was expecting a lynch mob" but before long took his coat off and stayed a while. He stated emphatically that the Rainbow would not be folded into the PCM, and that they would cover the various new computers - when available to the general public. He mentioned that the plans are on for the Chicago fest next year, and that Dale Pucket shelved work on his book "An Affectionate History of the CoCo" due to lack of time. The most interesting comment (I thought) was that people had been complaining about an increase in OS9 coverage in the magazine.

And ending out Saturday's seminars was Kevin Darling with his usuall Q and A into the wee hours of the night. Actually, it was only about four hours long (yikes!!). About the infamous Level 2 upgrade, Kevin said that when Tandy dropped the CoCo, they dropped the upgrade project, a collaborative effort between Tandy, MW, and the Darling gang. Officially, it doesn't exist, never did, never will. Although we know better, convincing the powers that be that there enough interest left in the CoCo market to do so much as even think about it is the problem at hand. Evidently, there has also been a recent change of management, and the new guy had to be educated.

Kevin mentioned several possibilities for making the upgrade yet available, including getting some people together to pay Tandy a royalty and sell it, or maybe if everybody chipped in so much we could outright buy it? Also stated was that the delays in getting it were partially his "fault, we kept putting in more and more [good] stuff". Tandy had given free reign on what to add, but made the restriction that it continue to run on a 128k machine - which also slowed development.

The upgrade features overlapping and resizable windows, as well as the capability to move a window to another 'screen' via the @. Kevin listed the more than half dozen people who worked on the project and commented on how well they had all kept the secret during development.

Kevin also spoke about a new publicity campaign for CD-I to get underway soon. He thinks it may very well boost OS9 awareness and help the rest of us out. There are a lot of OS9 usage out there that people just aren't aware of. For instance, did you know that the cashier computers with the scale on top in U.S. Post Offices run on OS9?

At one point Kevin stopped and asked anybody who was not a OS9 user to raise their hand - and only one person did! And only one other person had a CoCo3 with only 128k! Also discovered was that three quarters of those attending access one network or another.

Sunday afternoon brought Frank Hogg, Paul Ward, and Ed Gresick (Delmar) together to butt heads over their differing machines. Frank spoke of four major design changes in the TC9. A full meg of ram can be put on the motherboard directly with simm's rather than using an upgrade. Frank also added an interrupt in place of the $FF00 and $FF20 pia chips (the only differences realistically between the TC9 and CoCo3) allowing software emulation of their original function. What does all this mean? All OS9 software, Game Paks, and RSDOS software will work without change! A definate improvement and worth the wait. Frank admits that some things may run slower, and there will always be that one rogue program that messes it up, but the majority of existing software will now run just fine on the TC9.

Steve Bjork gave the last seminar on game programming, but with some extra time mentioned an interesting device he is selling in another market. His "Mileage_Accountant System" is a little black box that hooks up to your car and automatically logs mileage charged to various account numbers. It can be removed and plugged into a computer to dump the log in a database.

All in all, the fest was a wonderful success. Although the facilities were a little smaller, it seemed as crowded or more so than Chicago Fests gone by. And most importantly, it was a heck of a lot cheaper - rooms were about $35, and even the cost to the exhibitors was economical. There was plenty of new software to check out, and seminars to go to (not all were listed here). Although the exhibitors seemed to be half RSDOS and half OS9, OS9 was definately the majority interest of the people attending. So I think that although they may want to rename it the Atlanta OS9-Fest after a few years, it will definately be a regular occurance. And hey, we didn't have to listen to Lonnie sing.

Of course the best thing of any fest is getting to talk with people, of which I certainly did my share. I always get the worst raspiness to my voice during one of these things. I thank all of you who spotted my name (not hard to do eh?) and complimented me on this mag, it's always a boon to know somebody out there appreciates your hard work. And although the show was over at 4pm, I didn't get out of Atlanta till 9 due to a Chinese dinner with a Proctor or two... ;->

BBS_List

BBS_List

Send additions, corrections, and deletions to:

the OSKer
P.O. Box 24285
Speedway IN 46224

or call (317) 241-6401, or mail to:

sysop@root (StG-net)
StG@hummer.iupui.edu (I-net)
72427,335 (CIS-net)

Phone	Name	Max Bd	Net	Sysop
201-494-3649	MicroFone TBBS	9600		
201-967-1061	R.C.I.S. Hq.	2400	RCIS	
203-399-7394	CoCo Kingdom	1200		
203-444-1597	One-Byte	2400		
204-582-3593	Alpha SOftware	2400		
205-598-2100	Access CoCo	1200		
206-285-8335	Farpoint Station	2400		
206-425-5804	Columbia Hts	2400		
213-461-7948	Zog's Cavern BBS (ZOG)	2400	StG	Alan Sheltra
215-375-8814	Garf's BBS	2400		
215-376-1819	Glass Menagerie	2400		
301-863-5312	CoCo County Airport II	2400		
303-343-6707	Ribbs Hq.	2400	Fido	
303-757-6197	USS liberty	2400		
306-373-5029	Computer Link	1200		
312-745-1387	CoCo Rame	2400		
313-268-0028	Mega CoCo	2400		
313-879-2318	Arthur BBS	2400		
317-244-3159	Indy OS9 (ROOT)	2400	StG	Scott Griepentrog
319-362-6577	Color Connection	2400		
403-320-6481	Anarchy Intl' BBS (AIB)	2400	StG	Corey Lee
403-329-6438	So. Alberta BBS (SAB)	2400	StG	Dieter Rossman
404-446-6336	Online With Hayes	9600		Kent Pirkle
404-565-2977	ACCUG BBS	2400	Fido	Phil Zelgler
404-951-1540	INDEX System	9600	Fido	Rodney Alda
405-752-8955	Micro-Link	2400		
405-755-0768	CoCo Net III	2400		
407-433-3387	CoCo Zone	1200		
413-447-2346	Sun BBS	1200		
419-829-4825	Colorama PBBS	2400		
501-661-0527	Workshop (WORKSHOP)	2400	StG	Al Fleagle
504-649-5761	Alpha Software	2400		
508-792-0381	Gravyard BBS	2400		
515-432-7853	The Tomb	2400		
516-795-5874	Long Island CoCo Club	9600		
518-372-2694	Fishline BBS	2400		
519-753-7420	Multi-Net BBS	2400		
602-844-7840	The Pub	2400		
608-655-3806	Tandy Users Group	2400		
612-780-8936	TCOS9UG BBS	2400		
614-446-7430	Rainbows End (nightly)	2400		
614-965-1527	LastOutpost BBS	2400		
615-265-2629	Old Folks Home	9600		
615-383-0727	Nashville Exchange	2400		
616-684-1283	Zone BBS	2400		
617-661-0776	Firehouse	2400		
619-571-6366	8-bit Tandy	2400		
703-323-7654	PBBS	2400		
708-352-0948	S and V BBS (SANDV)	2400	StG	Paul Jerkatis
714-781-5825	Pegagus	2400	RCIS	Mike Jones
714-831-6530	Color Galaxy Milky Way	2400	RCIS	Eric Levinson
717-266-1935	Hacker's Haven	2400	Ribbs	Denny Hoin
717-393-5195	Citadel	2400	Ribbs	Mike Nelson
717-394-3654	Warlock's Cave	2400	Ribbs	George Felber
717-741-5078	Penetrator	2400	Fido	Lemar Whipperman
719-635-7228	Time Safari	1200		
804-744-9260	Tree House BBS	2400		
818-772-8890	Plain Rap	2400		
902-634-3095	Color Nova	2400		
904-245-6585	Nobody's Home (HOME)	2400	StG	Scott Proctor
904-595-2184	Snoopy's DogHouse (GIJOE)	2400	StG	John Swinson
915-682-6420	Phantom Mesa	2400		
916-392-8080	CoCo Motion	2400		

Fiction

FROM: Wayne@Zog (Wayne Campbell)
AREA: coco
SUBJ: FICTION.FICTION
DATE: 90/08/26 15:04:04
RCVD: root 90/08/27 00:37:07

The story you are about to read is NOT true. No names have been changed, because there's noone to protect.

I was walking down a street near my home the other day, when I chanced to see a piece of paper being blown in my direction by the cool breeze that always accompanies the setting of the evening sun.

When it reached me, I picked it up, and was amazed at what I read! I decided to post it here, in this message, so that others could also read it, and draw their own conclusions.

I have already formulated my own opinions concerning it, and will respond to it following the body of the message.

Here is the complete text of the message:

I was sitting at my desk, considering my beloved OS9, thinking of all that it can do. Though I am a novice, hardly beginning to understand the internal workings of the operating system, and having borrowed a copy of the Masterpiece, Inside OS9, written the the MASTER Kevin Darling, the GURU of OS9, (though his Genius is still too much for me to grasp), I had begun to consider the awesome possibilities that lay ahead in the future of this magnificent operating system, when it occurred to me that there were SHORTCOMINGS in OS9 that couldn't be cured except by a total rewrite of the KERNAL, and possibly a few other system modules! (What HORROR! My beloved OS9 WASN'T PERFECT!!)

I was in SHOCK! And, just when I thought I'd been stabbed through the heart, another DISASTROUS TRUTH ocurred to me! Even with a rewrite of OS9, the limitations of the 6809 COULD NEVER be changed! WHAT PAIN! WHAT ANGUISH!! The beginning of the END of OS9! To be sure! But, WAIT!!!

OSK!!! (OS9-68000, to those who don't know.) THERE was the answer! It had to be! The 680x0 processors were MUCH more powerful and AGILE than the 6809 series! And NO UNMANAGEABLE MEMORY LIMITATION!!! UP TO 16 MEGS AT ONCE!!! I was elated! The answer was there! Now, what about the OSK?

Then, the letdown. OSK, in its current form would never do! No graphics, no windows, just unlimited possibilities. Yet, I'm sure I heard that THE GURU HimSelf was working on a version of OSK for a new machine that HAS windows AND graphics!! OH WONDERFUL DAY!! Perhaps HE would take care of the little things that would make OS9 what it was meant to be!!

I don't recall exactly what happened then. I may have fallen asleep, but I just don't know. I found myself at a convention of various operating systems developers, and was looking over the various displays that were placed all around the hall. I saw (YECH!) MS-DOS, OS/2, UNIX, and a few others I'd never heard of before, and the ones I WAS familiar with seemed to be being treated as though they were totally obsolete and outdated. (I wasn't sorry for them, except maybe for UNIX!) Yet, nowhere to be found was OS9 or OSK! Why? Surely they would be here! I noticed a calander on a far wall, and saw that I had, somehow, been brought into the future!! It was the year 2011. What happened to OS9???

I asked someone close to me where the Microware display was, and they said, 'Microware? They went out of business about 14 years ago! Where have you been?!?!'. I felt as though I'd been shot! Microware GONE?!?! 'What happened??', I demanded to know.

'Motorola decided to stop manufacturing their 680x0 series of processors, and started what was supposed to be a more powerful series. Microware wasn't able to get their OS to work with it.', I was told.

'WHAT?!?!', I shouted! 'OS9 could run on ANYTHING!!'

'Apparently not!', the man exclaimed.

What had happened? Surely OS9 hadn't DIED! I walked around asking others, and soon found someone who seemed to know the whole story.

It seems that when Motorola developed the new '99' series processors, they found that all conventional OS's were obsoleted by it. NOTHING would run on it. SUPERCOMPUTER-ON-A-CHIP was its nickname. UNIX was too bulky, even though the RAM availability for the chip was in the GIGABYTES! And UNIX also had too many problems keeping up with the system. OSK was tried out on it, but was found to lack in ability to adapt to the new hardware. Microware tried writing several new versions, but none worked, and the cost was so enormous that they soon had to stop all attempts. The new chip was soon taking over in all sectors, even though an adequate OS hadn't been developed. Software was being written in binary that would run, and accomplish its task, so the various users were able to use it without much problem. Even now, the new OS's I had seen were all attempts to create a usable OS for the new micro. It was the opinion of this man that NONE of them would work out!

'So, what happened to Microware and OS9?', I asked.

'When the markets started buying the new processors, the old OS's were done away with. Many of the users of OS9 no longer needed Microware, and they went out of business.', I was told.

What dismay! What could be done? The new hardware was so powerful it literally ATE UP any OS put into it! I asked around some more, and soon found the primary cause of the problem. The existing OS's required SO MUCH overhead that the processor literally REFUSED to accept it! Memory management capabilities were so LIMITED that the processor couldn't function properly. I/O was no problem, but even attempts to incorporate AI were unsuccessful in reducing overhead and creating better memory management capabilities.

Suddenly, I was in a different place. I still don't know how I got there! I was in a room, not very large, and all over the walls was the following phrase:

Operating SYSTEM   DISK Operating SYSTEM   DISK SYSTEM

What did it mean??? Of course, I know what an operating system is!! But, what does it mean??? I sat down on the shiny, white tile floor, and closed my eyes, and thought, 'What am I doing here?'.

I don't know how long I sat there, but some time later, I heard the sound of air, rushing as though through a venturi. I opened my eyes, and the room had changed. There was now a door in one of the walls, and on the door was a plaque that said:

ENVIRONMENT ROOM

'What is this?', I thought outloud. I opened the door, and saw equipment, running and working, as though to provide some service to an unknown being. I looked again at the plaque on the door, and I understood. This room was the heart of the environmental maintenance service, for whomever it maintained.

'But, what's this got to do with OPERATING SYSTEMS?!?!', I queried to noone!

Then I was back at my desk. I still haven't figured out what it meant, or why I was shown all of this. What did it have to do with me? Why is OS9 DOOMED? Why is Microware doomed? Isn't there anyone who knows the answer?

First, I'd like to respond to the issue of Kevin Darling. While I'll be the first to admit that he knows far more than I, and possibly all of us, about OS9, I doubt that he has achieved 'GOD' status for his achievements. Anyone who would place him on such a pedestal should seriously consider returning to reality!

Second, the issue of OS9 vs. Other OS's. I agree that OS9/OSK are by far superior to any other OS I've seen, the idea that the knoweledge of its imperfection is cause to be harmed, and that the shortcomings of both the OS and the hardware it was designed for are cause to fortell the demise of the OS, is just a bit absurd!

Also, while the other OS's may be inferior overall to OS9, that, by no means, proves that they do NOT have their good attributes, NOR that they cannot be improved!

Third, the issue of new technology. While 'Only God Knows The Future', it is not unreasonable to think that a new technology could be developed that could obsolete everything currently in use. However, I doubt that the 'Market' would ever opt for a system with no controlling system to it. And the manufacturer of the technology surely wouldn't release it without one.

However, if such a thing DID ever happen, then it is possible that OS9/OSK and EVERY OTHER OS in existence would perish. Not just ONE!

Supercomputer-on-a-chip? Maybe so...stranger things have happened! I would think that it would present some interesting problems for the OS developers, but nothing so drastic that it couldn't be worked out, and, MAYBE, OS9/OSK would be the OS that SUCCEEDED, NOT FAILED!!

Fourth, the issue of Operating Systems. It is well known that the primary purpose of an operating system is to manage disk I/O. That is what they were developed for. However, I see the time coming when an operating system will have to do MUCH more than just manage I/O! I see the time when memory resources will need greater management control than is currently being applied. OS9/OSK, in my opinion, does a much better job of this than any other OS I've seen. Yet, even this will have to be improved as new technology is developed. Maybe we SHOULD start working on this now, and not wait for the technology to arrive, but it will have to be the decision of those better qualified than I to do or not to do.

Perhaps the idea of an OPERATING ENVIRONMENT needs to be thought out more completely than is currently being done in the concepts of DISK OPERATING SYSTEMS. (And, maybe, that is what the intent of the man's experience was!)

Last, I don't have any idea what it has to do with the author of the message!!!

Also, where has this guy been for the last year?!?! He's ONLY JUST HEARD of the MM1 and TC9/70 computers, and KD's work on OSK?!?!

I hope you've enjoyed reading this little stretch of the imagination. I enjoyed writing it! If it was a bit lengthy, I apologise.

```
--
+------------------------------------------------------------------+
|   Wayne@ZOG    |      OS9 IS SO MUCH FUN!!      |    Memphis@ZOG   |
|------------------------------------------------------------------|
|                                                                  |
| KIRK TO ENTERPRISE! BEAM ME UP SCOTTY! I'M WAITING, MR. SCOTT!!! |
|                                                                  |
+------------------------------------------------------------------+
```

HACKER CONTEST #1

In the first issue, I put forth a challenge to write a fast random number generator. Well, after dredging through the deluge of programs, the results are in. And I have to say how impressed I am with the quality of programmers out there! We've really got some skilled players on our team!

All the entries received were judged on the basis of:

1) Outputs numbers 1 to 10 randomly, using each once.
2) Speed
3) Simplicity of code
4) A really neat hack

The Grand Prize is a whole year of the OSKer for free, and the authors of all other entries printed receive 6 months free.

HONORABLE MENTION

Keith Alphonso sent in this very interesting solution in the form of a recursive program. While it does output numbers 1 through 10, it's not 100% random. But it is a neat hack - I wouldn't have thought of doing it this way.

```
#include <stdio.h>
#include <stdlib.h>

#define random(x) (rand(32767)%(x))

main()
{
   ranprint(0,101);
}

ranprint(min,max)
{
   int t;

   if ((max - min) < 2)
      return;

   /* Use the krieder random */
   t = random((max - min) - 1) + (min + 1);
   printf("%d\n",t);
   ranprint(t,max);
   ranprint(min,t);
}
```

THIRD PLACE

This program fits the bill by pre-setting an array, then stirring it around and outputing the numbers. Well structured and commented, but could be hacked simpler...

```
/* myrand.c
 *
 * My very own entry to The OSKer Hacker Contest.
 *
 * Algorithm: The numbers 1 through MAX are stored in an array. Then
 * the program simply loops through the array and swaps the current
 * element with some other, random element, thus scrambling the array.
 * Kind of like an 'unsort' routine! <grin>
 *
 * This could probably be trimmed down some, perhaps by only doing
 * MAX-1 swaps, but I decided to be thorough.
 *
 *         Ernest M. Campbell III
 *         campbell@ducvax.auburn.edu
 *
 */

#include <stdio.h>
#define MAX 10

/*
 * rnd(x), returs 0-(x-1) pseudo-randomly
 * this random routine was blatantly stolen from the
 * example in The OSKer ;)
 *
 */

int rnd(x)
int x;
{
   static long seed=25849;

   seed*=7;
   seed+=3;
   if (seed<0) seed= -seed;
   return (seed%x);
}

/*
 * swap(arrayptr, x, y), swaps elements x and y of the
 * array pointed to by arrayptr.  This routine is pretty
 * straightforward.  -- EMC3
 *
 */

swap(ptr,n,m)
int *ptr;
int n,m;
{
   int x;          /* temp variable used in swap       */
   x=ptr[n];       /* store first element for later use */
   ptr[n]=ptr[m];  /* one part of the swap....          */
   ptr[m]=x;       /* ....second part, and done.        */
}

/* Global array.  I could have moved this inside main. */
int array[MAX];

main()
{
   int i,r;

   /* Assign then numbers 1 through MAX to the array.  */
   for (i=0;i<MAX;i++) array[i]=i+1;
```

```
    /* Do the swap-thing!  Mash those elements around!  */
    for (i=0;i<MAX;i++) swap(array, i, rnd(MAX));

    /* Print 'em out!  We're done scrambling the array! */
    for (i=0;i<MAX;i++) printf("%d\n",array[i]);
}
```

SECOND PLACE

The next winner has written a simple two-loop routine. The first pass sets up the array like the first program, the second one pulls a random number out of the table and then sneaks it out of the list. This method is fast, simple, and a fairly neat hack.

```
PROCEDURE RANDOM
 0000       (* HACKER CONTEST ENTRY
 0017       (* WRITTEN BY PAUL JACKSON
 0031       (* USING BASIC-09
 0042       (* THIS PROGRAM PRINTS N RANDOM NUMBERS ONCE
 006E       (* NOTE THAT DIFFERENT HANDLING OF THE RND FUNCTION
 00A1       (* ALSO NOTE THAT THE MAXIMUM ARRAY SIZE IS 1000
 00D1       (* CHANGE THIS FOR MORE OR LESS MEMORY
 00F7       DIM N,A,B:INTEGER
 0106       DIM NUM(1000)
 0110       INPUT N
 0115       FOR X=1 TO N
 0128         NUM(X)=X
 0135       NEXT X
 0140 1     IF N>0 THEN
 014F         B=INT(RND(N-1))+1
 0162         A=NUM(B)
 016E         PRINT A; " ";
 0178         NUM(B)=NUM(N)
 0187         N=N-1
 0192         GOTO 1
 0196       ENDIF
 0198       PRINT
```

FIRST PLACE

The title of KING HACKER goes to Edgar Zuman (who lives in Holland) for his neat hack. This program is as good as it gets - one loop runs the whole routine, which is just a few lines of code!

```
/*
 * rnd(x), returns 0-(x-1) pseudo-randomly
 * this random routine really sucks, but is just an example
 * in case your library doesn't have one
*/
rnd(x)
int x;
{
    static long seed=25849;

    seed*=7;
    seed+=3;
    if (seed<0) seed = -seed;
    return(seed%x);
}

#define MAX 10

int bucket[MAX];

main()
{
    int i,r;

    for (i=0; i<MAX; ++i) {
        r = rnd(MAX-i) + i;
        printf("%d\n", bucket[r]?bucket[r]:r+1 );
        bucket[r] = bucket[i]?bucket[i]:i+1;
    }
}
```

It just looks way too short to get the job done doesn't it? That's one of the signs of a really well done hack. Don't believe it works? Key it in an try it!

THE NEXT CHALLENGE

Because of the time it takes for people to come up with a neat hack and all the neat ideas I keep coming up with and can't decide between, a new contest problem will be issued each month. Results will be published after a few months. Make sure to identify which contest number you are going for when sending in your entries.

HACKER CONTEST #2

Write a random number generator (C code preferred), as a replacement for the very poor one used as an example in the first hacker contest. The function should be passed a number N and return a number (int) between 0 and N-1. It should seed itself the first time it is called by making a call to a function like getime(). The winner will be selected on a basis of 1) randomness factor, 2) speed, 3) small, effecient code, and of course, 4) hack-ness.

The program listed below (with the old junky rnd() routine as an example) will be used to determine how well it calculates random numbers. The number of times each number comes up is counted and how far off it is from the ideal is printed at the end. Use it to test your own routine - the closer to 0, the better it is. The horribly bad example routine (which I only took a few minutes to write originally) finished with an unbelievable 187% off par - surely it should be easy to improve on that. WARNING! This program takes a LONG time to run! About a day, in fact, on a 68k machine. To increase speed decrease the LOP setting to a few hundred.

```
/*
 * rnd(x), returns 0-(x-1) pseudo-randomly
*/
rnd(x)
int x;
{
    static long seed=25849;

    seed*=7;
    seed+=3;
    if (seed<0) seed = -seed;
    return(seed%x);
}

#define MAX 15000  /* random spread */
#define LOP 10000  /* count should be close to */

int temp,worst;
unsigned int count[MAX];

/*
 * random number tester
*/
main()
{
    register long loop;

    /* count up occurances of each number 0-MAX */
    loop=MAX*LOP;
    while (loop--) count[rnd(MAX)]++;

    /* find worst off */
    worst=0;
    loop=0;
    while (loop<MAX)
    {
        temp=count[loop]-LOP;
        if (temp<0) temp=-temp;
        if (temp>worst) worst=temp;
        loop++;
    }

    printf("Max deviance: %d  %d%%\n",worst,worst*100/LOP);
 }
```

O_Say_Can_You_C

By Al Fleagle

Well, here it is. Yet another article in yet another magazine promising to teach you the wonders and workings of the C programming language. If you're like me you've read a dozen such articles and you still don't know Kernighan from Ritchie.

So what are you going to need? I'm using a Color Computer with 512K of memory, a 360K floppy, a 40meg hard drive, and an Epson printer. I don't intend to describe how to get the C compiler to work on two floppies. Just take my word for it, it can't be done. It has never been done. And anyone who has done it would mow his lawn with a pair of scissors.

In addition to those items listed above, you will need a copy of the C Compiler. Right now there are great buys at many of your local Radio Shack stores, so if you don't have a copy, go pester the local Tandy manager who probably doesn't even know the Color Computer has a C Compiler. Do not order a copy until you have looked in every Radio Shack you ever hope to visit. No sense in paying the full $99.00 price if you can find it for $20.00.

Get the two disks from the C package. One disk is labeled "C Compiler" and the second is labeled "C Library". Begin by placing the "C Library" disk in your floppy drive. If you type 'dir /d0' (substitute the name of your floppy in place of "d0" if it is different) you will see a directory of d0.

```
DEFS          LIB          SOURCES
```

We must copy the files contained in these directories to our hard disks. Of course those of you with lots of OS9 experience will know easier ways to copy the files to hard disk. My favorite is:

```
dircopy /d0 /dd D        (* respond 'Y' to any rewrite prompts *)
```

That's it, you've completed the transfer of all the files on the C Library disk. (If anyone would like a copy of dircopy, mail me a note asking for dircopy. Please include $4.00 for shipping and handling. My address is at the end of this article.)

If you don't have dircopy, type the following:

```
makdir /dd/DEFS
makdir /dd/LIB
makdir /dd/SOURCES
```

This will make the directories needed on your hard drive. Now let's look at the DEFS directory. Type 'dir /d0/defs' and you should see:

```
ctype.h      direct.h     errno.h      modes.h      module.h
os9.h        os9defs.a    setjmp.h     sgstat.h     signal.h
stdio.h      time.h
```

If you used the dircopy command, you're already done. Otherwise, to transfer these files to your hard disk, type the following:

```
copy /d0/defs/ctype.h  /dd/defs/ctype.h
copy /d0/defs/direct.h  /dd/defs/direct.h
copy /d0/defs/errno.h  /dd/defs/errno.h
copy /d0/defs/modes.h  /dd/defs/modes.h
copy /d0/defs/module.h  /dd/defs/module.h
copy /d0/defs/os9.h  /dd/defs/os9.h
copy /d0/defs/os9defs.a  /dd/defs/os9defs.a
copy /d0/defs/setjmp.h  /dd/defs/setjmp.h
copy /d0/defs/sgstat.h  /dd/defs/sgstat.h
copy /d0/defs/signal.h  /dd/defs/signal.h
copy /d0/defs/stdio.h  /dd/defs/stdio.h
copy /d0/defs/time.h  /dd/defs/time.h
```

Now we need to transfer the LIB directory. Again, if you used dircopy you're already done. Otherwise, type the following:

```
copy /d0/lib/clib.l  /dd/lib/clib.l
copy /d0/lib/cstart.r  /dd/lib/cstart.r
```

And finally we must transfer the SOURCES directory to the hard disk. Dircopy users, go watch some TV. The rest of you, type the following:

```
copy /d0/sources/line.c  /dd/sources/line.c
copy /d0/sources/prof.c  /dd/sources/prof.c
copy /d0/sources/rdump.c  /dd/sources/rdump.c
makdir /dd/sources/SYS
copy /d0/sources/sys/abort.a  /dd/sources/sys/abort.a
copy /d0/sources/sys/access.a  /dd/sources/sys/access.a
copy /d0/sources/sys/cfinish.a  /dd/sources/sys/cfinish.a
copy /d0/sources/sys/change.a  /dd/sources/sys/change.a
copy /d0/sources/sys/comp.sys  /dd/sources/sys/comp.sys
copy /d0/sources/sys/cstart.a  /dd/sources/sys/cstart.a
copy /d0/sources/sys/dir.a  /dd/sources/sys/dir.a
copy /d0/sources/sys/id.a  /dd/sources/sys/id.a
copy /d0/sources/sys/intercept.a  /dd/sources/sys/intercept.a
copy /d0/sources/sys/io.a  /dd/sources/sys/io.a
copy /d0/sources/sys/make.sys  /dd/sources/sys/make.sys
copy /d0/sources/sys/mem.a  /dd/sources/sys/mem.a
copy /d0/sources/sys/misc.a  /dd/sources/sys/misc.a
copy /d0/sources/sys/mod.a  /dd/sources/sys/mod.a
copy /d0/sources/sys/process.a  /dd/sources/sys/process.a
copy /d0/sources/sys/profdummy.a  /dd/sources/sys/profdummy.a
copy /d0/sources/sys/signal.a  /dd/sources/sys/signal.a
copy /d0/sources/sys/stat/a  /dd/sources/sys/stat/a
copy /d0/sources/sys/syscall.a  /dd/sources/sys/syscall.a
copy /d0/sources/sys/syscommon.a  /dd/sources/sys/syscommon.a
copy /d0/sources/sys/tidyup.a  /dd/sources/sys/tidyup.a
copy /d0/sources/sys/time.a  /dd/sources/sys/time.a
```

(Dircopy for $4.00 sounds like a bargain right now, doesn't it?)

Now replace the C Library disk in your floppy drive with the C Compiler disk. Type 'dir /d0' and you should see:

```
CMDS
```

Let's see what's inside the CMDS directory on the C Compiler disk. Type 'dir /d0/cmds' and you should see the following:

```
c.asm        c.link       c.opt        c.pass1      c.pass2
c.prep       cc1          copy         del          dir
echo         list
```

Some of these programs should look familiar such as copy, del, dir, echo and list. You should already have a copy of each of these in the CMDS directory of your hard disk so there is no need to copy them again. Instead copy the remaining programs to your '/dd/CMDS' directory. Assuming your floppy is '/d0' and your hard drive is '/dd', type the following:

```
copy /d0/cmds/c.asm /dd/cmds/c.asm
copy /d0/cmds/c.link /dd/cmds/c.link
copy /d0/cmds/c.opt /dd/cmds/c.opt
copy /d0/cmds/c.pass1 /dd/cmds/c.pass1
copy /d0/cmds/c.pass2 /dd/cmds/c.pass2
```

You will notice we didn't copy the 'ccl' program nor the 'c.prep' program. These programs won't work on a hard disk. The programs 'ccl' and 'c.prep' are hard-coded to use the floppy '/d1' when looking for libraries and such. We need to change these programs to look on '/dd'.

For 'ccl' the changes we need to make are:

Offset Address	Old Value	New Value
00000EE6	31	64
00001F4F	3C	21
00001F50	A4	24
00001F51	DF	D4

All this really amounts to is changing a '1' at offset address EE6 to a 'd' and verifying the CRC. CRC is the last three bytes in a file, and stands for Cyclic Redundancy Check. (Translation: Black Magic)

To make these changes I am assuming everyone has a copy of modpatch which comes with OS9 Level II. To use it you need to type the following:

```
chd /dd;build ccl.patch
```

You will see a '?' at the start of each line. Type each response as indicated. You should press the enter key wherever you see <enter>.

Prompt (Do not type this)	Response (Type this)
?	L ccl<enter>
?	C 0EE6 31 64<enter>
?	V<enter>
?	<enter>

Next run modpatch by typing the following line:

```
load /d0/cmds/ccl;modpatch ccl.patch;save /dd/cmds/ccl ccl
```

Now you should have a patched version of ccl in your /dd/CMDS directory. To be sure the patch was sucessful, type the following:

```
cmp /d0/cmds/ccl /dd/cmds/ccl
```

You should see the following:

```
 Differences

byte      #1 #2
========  == ==
00000EE6  31 64
00001F4F  3C 21
00001F50  A4 24
00001F51  DF D4

Bytes compared:   00001F52
Bytes different:  00000004
```

If you get something different, rewind and try again.

Now we need to make the patches to the c.prep program. It is the same process as we used for the ccl program. You need to type the following:

```
build cprep.patch
```

You will see a '?' at the start of each line. Type each response as indicated. Again, press the enter key wherever you see <enter>.

Prompt (Do not type this)	Response (Type this)
?	L c.prep<enter>
?	C 135D 31 64<enter>
?	V<enter>
?	<enter>

Next run modpatch by typing the following line:

```
load /d0/cmds/c.prep;modpatch cprep.patch;save /dd/cmds/c.prep c.prep
```

Now you should have a patched version of c.prep in your /dd/CMDS directory. To be sure the patch was sucessful, type the following:

```
cmp /d0/cmds/c.prep /dd/cmds/c.prep
```

You should see the following:

```
 Differences

byte      #1 #2
========  == ==
0000135D  31 64
000027A9  2A 4B
000027AA  E9 70
000027AB  AD BB

Bytes compared:   000027AC
Bytes different:  00000004
```

And now for the final test. If everything has gone correctly you should be able to type the following:

```
chd /dd/sources;ccl line.c
```

Wait for the compiler to compile, the computer to compute and the floppy to flop. Now type 'dir /dd/cmds' and somewhere in that mess you should find a new program called 'line'. To test this new program, type the following:

```
line line.c
```

You should get the following:

```
/* line:  list lines of file with numbers */

#include <stdio.h>

main(argc,argv)
char **argv;
{
    char line[256];
    register int count = 0;

    if(--argc)
        if(freopen(*++argv,"r",stdin) == NULL) {
            fprintf(stderr,"can't open file: %s\n",*argv);
            exit(errno);
        }

    while(gets(line))
        printf("%5d %s\n",++count,line);
}
```

PLACE

AD

HERE

If you compare this against the original source code, line.c, you will notice that numbers have been added before each line. This is what the line program does, it lists text files and numbers the lines.

Well, you are rapidly on your way to becoming an expert C programmer. Try compiling the other source codes in the SOURCES directory, prof.c and rdump.c. Then you can type in the source codes listed elsewhere in the OSKer magazine and try compiling them. Don't be disappointed if they don't work because they may require libraries you don't yet have. But we'll get there soon enough.

Next time we might look at installing the RMA in place of ASM in your C Compiler, possibly improve the cc1 program further, perhaps discuss what all these libraries and programs actually do and how to upgrade them, and hopefully begin translating source code into English.

Meanwhile, if you have problems or questions, would like a copy of the dircopy program, or would like to make suggestions for this column, please contact me at the following addresses:

Al Fleagle
11 Alpine Court
Little Rock, AR 72205
501/661-1063 voice

501/661-0527 sysop@WorkShop, StG Network
(24 hours, unless I'm hacking my hardware)

CIS 72527,1354

Playing_Chess_in_C

Installment #2, by Scott Griepentrog

Okay, did everybody spot that bug in the first (last month's) part of the chess program? It's in the setbrd() function. Did you run it? Did you see what's wrong with it? Well; while you're looking for it I'll repeat the last paragraph the first installment:

And if you get bored in the meantime, see if you can figure out a way to condense (simplify) the setbrd() function. I can see a number of things that would bring the size of the function down to about 10 lines or less even.

So how does one go about simplifying C code? Well, it's much like simplifying algebraic expressions actually (uck!). But it can be turned into a game of finding out how many different ways you can - once you learn the few basic rules: (1) Look for patterns that can be exploited, (2) use more advanced features of the language. Let's take a look at setbrd() one section at a time:

```
    /* first row */
    board[0][0]='R';
    board[0][1]='N';
    board[0][2]='B';
    board[0][3]='Q';
    board[0][4]='K';
    board[0][5]='B';
    board[0][6]='N';
    board[0][7]='R';
```

Using a crazy nutzoid feature of C, we can cut three lines out of the above really easily. Anytime you do x=y; or x=y*3+5; or x+=y; or really _any_ expression, it actually returns a value. You don't see it, you may not do anything with it, but it's there. For example, you can say putchar(x=32);. What does this do? First, assign x to a decimal 32 (space in ASCII), second, pass the value assigned to x on to putchar() which outputs the space character. Why would you want to do something as confusing as that when x=32; putchar(x); would be easier? For one, it takes less space. If you make a habit of doing this sort of thing you could very easily cut the length of your source code in half. Which means less code to look through when looking for a bug. It also helps the speed of the program to some degree.

So how can we use this principal? If one can say x=32; and x=y; then one should be able to do x=y=32; right? See where I'm headed yet? See the three lines we can eliminate:

```
    /* first row */
    board[0][0]=board[0][7]='R';
    board[0][1]=board[0][6]='N';
    board[0][2]=board[0][5]='B';
    board[0][3]='Q';
    board[0][4]='K';
```

Still, there are other methods to simplifying the same routine. What about another approach altogether. How else can we accomplish the same thing without doing it the same way? That's the real challenge - and the real fun when trying to out-hack the next guy. Actually, that's pretty much how hacking got started - a bunch of brainy types sitting around this machine trying to fit more and more program into the same one thousand bytes of memory. It was a big contest to see who could do the best code crunching. But then, that was in machine language, and I've digressed. How about this:

```
    int x;
    char *royalty="RNBQKBNR";

    x=0;
    while (x<8)
    {
        board[0][x]=royalty[x];
        x++;
    }
```

Now that is a different way to do it. Set up the first row as a string of characters, and then go through a loop to set them in place. But it's not all that shorter, in fact, it isn't really shorter at all. So why use it you ask. Do you get the feeling I've suckered you into that question? Okay, the answer is actually found by looking at the code that does the last row. Looks sorta familiar eh? Well, it should. Notice that everything is the same except that it is in lower case. Hmm... I see a light bulb hanging over your head. There's the tolower() function, isn't there! Since we're already going through the row with the first set of pieces, why not kill two rows with one loop? By adding only one line, we get a function that replaces both parts and it shorter to boot.

```
        board[7][x]=tolower(royalty[x]);
```

That worked pretty well! I wonder - what about the rows of pawns?

```
        board[1][x]='P';
        board[6][x]='p';
```

Cool beans! That just got rid of four fifths of the code! Now, the only routine left is:

```
    /* 6 empty rows */
    y=2;
    while (y<6)
    {
        x=0;
        while (x<8)
        {
            board[y][x]=0;
            x++;
        }
        y++;
    }
```

Which is also easy to cut down. Rather than add another loop to set the remaining six rows, we can use that assignment trick we learned earlier:

```
        board[2][x]=board[3][x]=board[4][x]=board[5][x]=0;
```

Which only takes up one more line! Now the whole setbrd() function is:

```
setbrd()
{
    int x;
    char *royalty="RNBQKBNR";
    x=0;
    while (x<8)
    {
        board[0][x]=royalty[x];
        board[1][x]='P';
        board[2][x]=board[3][x]=board[4][x]=board[5][x]=0;
        board[6][x]='p';
        board[7][x]=tolower(royalty[x]);
        x++;
    }
}
```

Now isn't that just so much shorter? And easier to read yet. A major improvement. Learning to simplify code in this manner is almost an art form at times - but it's really not difficult, providing you know the options open to you. A good knowledge of the language and a little imagination is all that is needed.

So, did you find that bug yet? It really wasn't so much of a bug as just a minor mistake - could have been a typo, but I know better. The line of code that sets pawns into the row index 6 should use a lower case p, not the upper case P. Rows 6 and 7 are in lower case to separate who's pieces are who's. Could get sorta confusing otherwise, no?

Okay, your assignment for next month is to... naw, sounds too much like a teacher. How about this - you simplify the dspbrd() function and send it in. The person with the best version gets their function made part of the 'official' code, printed in the mag, and the usual 6 months free. How's that sound? Gee, school sure would have been different if they gave away free stuff to the best students...

Okay, where wuz I. Oh yeah, about to start into a new piece of code. I know some of you have been standing out by your mailbox waiting for this one. Well, what to do next. Hmm, what do we have. Lesse, a routine that sets up the board for play, and another that displays the current settings. Well, sounds like it's time to move a piece, right? And just how do we tell our move function how we want to move? Since the board is already in x and y coordiates, that would probably be the best way. Of course, we can input from the player in a letter/number system, then translate to our numbers for processing. So what does this move function do really? Just move a piece? We can do that easily enough just by:

```
    /* horsey jumps over pawns: 7,6 - 5,5 */
    board[5][5]=board[7][6];
    board[7][6]=0;
```

Well, that's too simple to bother with a whole function for. Might as well just write that into the main code. But a move should only be allowed if it's a valid move. Certain pieces move only certain ways. Yah, but how to tell a computer that, right? That's definately worth writing a function for. Maybe it returns one number for an okay move and another if that move isn't allowed. Hmm. Or maybe this routine should just check for a valid move - leave the actual moving to another routine (to which can be added stuff for graphics). Okay, how about valid(a,b,c,d) to check a move from a,b to c,d. Sounds kosher. If we make it return 1 for valid, 0 for non-valid, then when we use this routine it would be like: if (valid(a,b,c,d));, which would make for easy reading. Kool.

So, now how brown cow to test each different piece's type of move? Sounds like a job for switch-case. Instead of going: if this, then that; for each piece, the switch is given what we're testing (what kind of piece) and a case statement for each case is supplied. Confused right? Well, lemme just show you what I mean:

```
valid(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
    /* gotta have a piece to move */
    if (!board[x1][y1]) return(0);

    /* now check what piece */
    switch (tolower(board[x1][y1]))
    {
    case 'p':  /* pawn (little guys) */
    case 'r':  /* rook (castle) */
        break;
    case 'n':  /* knight (horsey) */
        if (abs(x1-x2)+abs(y1-y2)!=3) return(0);
        if (x1==x2 ¦¦ y1==y2) return(0);
        break;
    case 'b':  /* bishop */
    case 'q':  /* queen */
    case 'k':  /* king */
    default:   /* say what? */
        break;
    }

    return(1);  /* must be okay */
}
```

Although I've only put in the code for the case of a knight, you can see how simple a switch-case is. Ah, but what do all these break; commands do, right? Okay, the switch case is really nothing more than a smart goto - the break causes the program to jump to the brace ending the switch statement. If I didn't put in that first break; after the rook case, for example, the first two cases would drop down into the knight case. I haven't put a separate break in for each case yet, but that will be necessary to separate the code, once written in.

The toupper() allows this routine to work to test moves for either side. That's an important consideration - if you remember, one we planned from the start. Re-using the same routine for another purpose not only makes good sense seeing as it cuts down on program size, but it is a sign of a programmer who plans ahead. The better planned a project is from the start, the better the code will work, look, and feel when it's done. Okay, okay, end of lecture.

End of this installment too. I'm out of space. But you ask, what about all that gobledy-gook in the knight case? What does it mean? Or if you understand what it does, why does it work? Or does it? When you find out, how about letting me know?

In installment #3: completing valid() and asking the player for a move...

Ramblings

How did I ever convice myself this would be an easy job...

While typing this, I find myself at the Northlake Holiday Inn. It's 90/10/06 03:05, (that's AM) and in about five hours starts the first CoCo Fest in Atlanta, Georgia. Actually, I hadn't originally planned on attending this fest - I had another obligation, not to mention the ten hour drive to get here. But due to an illness in my family (although nothing serious [yet]), I have restlessly waken up to stare my first prototype MM1 (no 2nd board or hard drive, eeeuch) in its face, or rather, dual 1.44 Meg drives. I have a deadline to meet, and I'm eager with anticipation, hopeful to see the first few new version of the MM1. Reportedly due to FCC clearance and chip supplies, the much awaited release of the MM1 has not met IMS's own deadline of middle of September, but they are supposed to have several here for the fest.

Speaking of deadlines, most of you are probably wondering why a September issue would come out in October. Well, my dog ate the copy. Yah, right. Actually, I don't have a dog. But I do have a dog of a job trying to keep various client's computer systems running. And I've been spending a lot of time out of town writing software for some people who always want it done the week before I started. Nothing changes, I guess. But as busy as I get, please realize that nothing is as important to me as keeping this magazine running - and running consistantly, even if not quite on time at first. So, as soon as I get this issue to the printers, I'll be working on the next one right away. The more of these I do, the faster it will go. Or at least, so one would think.

To get these pages filled up and out to you faster, it also helps to have articles written by others than yours truly. You all don't know how much I appreciate all the articles that I have been sent (probably because I haven't had time to reply to you). I have enough to help fill the next few issues, but I can always use more. Don't be afraid to get down and dirty or technical or even outspoken - if someone choses to disagree, they can write a letter or article for inclusion as well. I promise to keep everybody abreast of all the OS9/OSK news as I hear it, but I've only got so many ears and BPU time (Brain Processing Unit). And no, my ears are roundy just like everybody else's.

So, here I sit, stuffing myself on fritos, milky ways, gatorade, and an apple, pumping Queen into my ears (yikes was that disk was expensive), typing on my new pint-sized XT keyboard plugged into my already outdated MM1 hooked to my EGA monitor with the handle for hauling around and thinking about taking a shower and cleaning up while wondering about the strange lyrics to "Bicycle Race", but yet sleepless over having a chance to rap with the people of CoCo/OS9/OSK'dom, who are better kind of people (in my opinion), much like Trekkies (I'm one of them too), but I think I had better hit the brakes and put in a period before I get stopped by the Grammer Police.

And now a word from our sponsors:

-+-

Sirs:

I would like to request that an extensive review of the MM/1 micro be available at the time of it's debut. The first issue (7/90) did not cover many essential components of the MM/1 specifications such as VSC graphics capabilities/limitations, video output (NTSC, 'true' multisync monitor required?), PC component connection capability (keyboard, power supply), CD-ROM support of different formats, etc...

Thanks, Jack W. Wine

Okay, here it is: video output of the MM1 is via a CoCo3 compatible 10-pin dual header connector. Any monitor that will work on a CoCo3 will work - but Multi-Sync monitors will add additional display capabilities. NTSC output (i.e. composite video like on a VCR) is not available right off the board that I know of, but creating an adapter is not out of the question. PC cases, power supplies, floppies, hard drives, and XT type keyboards can be used on the MM1. CD-ROM will be supported of course - stay tuned. See the last issue for description of the graphics capabilities. And stay tuned for a deep digging review of the full two board MM1 in an upcoming review - i.e. as soon as I get mine!

-+-

Dear Scott:

Glad to see that you are not afraid to "call it as you see it" with regards to Microware. Someone needs to let them know that even though they have a good product there is always room for improvement. It really infuriates me that they STILL have not fixed some utility bugs as of V2.2 that I reported existed in 1.2! I keep filling out PDR forms and sending them in with the hopes that someday the problems I report will be fixed.

I'd sure like to know what ever happened to 68 Micro Journal, they still owe me some issues! I posted a msg on CI$ and someone else had the same problem. Perhaps a bit of invenstigative reporting is in order?

Sincerely, James E. Truesdale

Yikes! I'm honestly suprised that Microware would be guilty of dragging their feet on a bug. They haven't gotten where they are by leaving bugs in their software. Ugh, maybe I should rephrase that... Anyways, how about sending copies of those problem report forms here to help those out there who might be beating their heads on the same bugs. I've heard that 68 Micro is not putting out any more issues, although I have not verified that yet. But I have talked with Ted at CoCo Clipboard - his magazine has folded. We've discussed sending OSKers in place of paid issues of the Clipboard.

-+-

Dear Mr. Griepentrog,

Thank You, Thank You, Thank You, for starting an OSK magazine. I have read the 68 Micro Journal but don't subscribe. I am still pretty new to OS-9 and desire a magazine to help me learn and to keep up with the goings on in the OS-9 world.

I really like the layout of your magazine. I don't agree with the general consensus that the Allocation Map and Attributes are "pushing it". When you get more authors it will be nice to have this information. I read almost every article in the first two issues. I am especially interested in learning programming, reading about new hardware and software projects, Doc_OSKer, interview, your rambles, Robert Miller's OS-9_in_Industry articles. There is so much information in the OSKer that I don't want to miss a single issue.

My only comment is that when you are talking about specific computers, ie: CoCo or 8 bit or 16 bit, if you could state it clearly?

A question: there is a hardware product called DC Port by Double Click Software which adds serial ports to the ST and I am wondering if it works with OS-9? I wrote them a letter but never got an answer.

Another question: I have an XT clone and you mentioned that you were using a PC as a terminal. I am wondering if I can I hook up mine via a serial cable and transfer binary and ARchived files from the PC's disk drives to the OSK's disk drives? I don't have a terminal program for OSK but I do for the PC.

Sincerely, David Nielsen

Well, gee, <blush>, I couldn't have said it better myself. Glad you like it! Okay, I propose that to clear up misunderstanding of which OS9 one is talking about that there be put the word COCO, 6809, or 68K in paren's after the first usage of OS9 in any article. Of course, OSK would mean OS9 (68k). COCO should be used to designate something that only works on that machine, while 6809 should be used for anything on that processor alone. If talking specifically about OS9 on an ST or MM1 or TC70 or TC9 or whatever, simply put that machine name in the parens, i.e. OS9 (ACME). If all versions of OS9 are to be talked about, use OS9 (ALL). And OS-9000 could be referenced as is or in the forms OS9 (C), OS9 (C-68k), OS9 (C-86). Okay everybody?

Although the serial board add-on for the ST you mention could be made to work under OS9 (ST), I highly doubt that the current drivers would work with it, that the manufacturer would supply OS9 drivers, or even that somebody has already written drivers for it themselves. But I could be wrong. If so, somebody please let me know.

The easiest way I have found to communicate between PC and OSK (ALL) is using any old terminal program (on either end) and Kermit for file transfers. I tend to use Procomm on the PC (under OSK the ANSI termcap entry can be used for full screen operation with programs like the umacs editor), which has a functional (barely) version of Kermit. Kermit is actually considerably slower than other protocols, but if you are running over a direct connection rather than phone lines this doesn't matter as much. It's advantage is that it automatically converts CR/LF correctly for text files, and it can handle a batch job of several files at once (use *'s on the PC or OSK, list multiple files on OS9 (6809)). Be warned, though, that Kermit defaults to TEXT transfers, and you MUST set BOTH sides to BINARY to keep it from trashing binary files. It's big disadvantage is that every implementation uses some different control keywords - you have to get used to asking for help all the time to get it right. And before you attempt to transfer an archive file from one machine to another, check to see if both programs have a program that reads the same archive format. Generally, PC's use .ARC and .ZIP, whereas OS9 (all) uses .AR and .PAK. Each type of archive is incompatible with the others. If you can obtain Kermit from a friend, simply start a shell on the serial port connected to the PC use your terminal program on the PC to poke around and call up Kermit for transfers. But start out with a safe baud rate like 2400 before trying faster speeds that OS9 (COCO) can't handle.

-+-

FROM: sysop@AIB (Corey (Chin) Lee)
AREA: stg/update
SUBJ: OS-9 update
DATE: 90/08/30 22:21:49
RCVD: SAB 90/08/30 22:22:15
RCVD: root 90/08/31 00:15:21

Hey gang, I was talking to an old friend Guy Loucks. He has informed me that the new update for OS-9 Level II should be out in about a months time!!!! Fun Fun Fun eh? The rumour is that it will be in PD!!!! Some person is just finishing the docs apparently. I guess 9 or so people have been in on this project. Many of the known bugs have been taken out and there are some enhancements also. The little IRQ hardware hack that some of you may have done to allow you to use 2 RS-232 paks, will have to be taken out since that was a hardware fix for a software problem. Gee, I'm glad I'm such a procrastinator. One neato thing is that if you have installed the DISTO 1 meg upgrade, you do not have to run mega to get it to recognize the extra 512k cuz it'll to it for you. So be on the lookout for it gang!!!!

And one question. I've also heard that the 68030 Tomcat will run at not quite 4 MIPS. Can anyone verify that? I've heard that a 25 Mhz 386 machine only runs at about 1.5 MIPS. If this is true, then the 68030 Tomcat will be a very good machine for the price. Not to mention that OSK will blow the doors off of MS-Youknowwhat. Nuff said.

```
---
|     Chin = Corey Lee = Sysop@AIB - Tree hugger and proud of it!!     |
|          Voice line = (403) 320-1443   (7:00 - 24:00 MDT)            |
|                       Happiness = OS-9 + StG                          |
---
```

Well, the word from the Kevin Darling is that the much waited for Level 2 upgrade is going to be a while yet - and that's still IF it comes out at all. Actually, this is a slight improvement as the project _was_ supposed to be shelved period. Evidently, there is now a slight chance that some deal to sell the package to the general public might be arranged. It would be interesting to know how many people would by a copy, at what price, and if that would add up to what the big T and the big M would want for it. Somebody could make a fair profit by buying up the project if the price was right. It all comes down to money, and quite frankly, I'm not holding my breath.

On the subject of the IRQ problem in the CoCo3, contrary to common rumor it IS a hardware problem. There are software patches, not fixes, that cut the problem to practically nil, but it can still cause the loss of a few characters here and there regardless of what driver is used with the ACIA. The diode fix is the simplist and most reliable way of solving the problem, although it does require opening the case (but no soldering).

There is no 68030 Tomcat. There are several K-BUS boards, including a TC9 that thinks it's a CoCo3, a TC70 that thinks it's an MM1, as well as 68000 and 68030 cards. The TC boards have a processor, memory, graphics, and I/O. But the 68k and '30 cards have only a processor, and thus are not part of the Tomcat series. Hey Frank, what's the word on your MIPS?

-+-

SYSTEM IV COMPUTER

PERFORMANCE - FLEXIBILITY - VERSATILITY

MC68000 Microprocessor - 16 MHz
1 MBytes of DRAM (0 wait state)
4 MBytes optional

VGA Video Card - 800 x 600 x 16
to 320 x 200 x 256
or - Hercules Monochrome Card
Seven IBM/XT Compatible slots
Floppy Disk Controller (37C65)
supports two 360K, 720K, 1.2M
or 1.4M Drives
One 1.4 MByte, 3 1/2" Floppy Drive
WD XTGEN Hard Disk Controller (MFM)
Four RS-232 Serial Ports
Parallel Printer Port
40 MB Hard Drive, 28 msec, optional
Clock with Battery
4 layer board
Memory Expansion slot - adds
up to 8 MB of DRAM.
AT style keyboard
200 Watt Switching Power Supply
Professional OS9/68000
Drivers and descriptors for the
devices and ports provided.
Baby AT Case - holds three 5 1/4"
1/2 ht drives and one 3 1/2"
drive accessable from the
front and one 3 1/2" internal
drive.
One year parts and labor warranty
Mfg by Peripheral Technology

Model K402-m with Hercules Monochrome Video Card	$1,399.00
Model K402-v with VGA Video Card	$1,499.00
Model K401-m with 4 MB on-board DRAM, 40 MB Hard Drive (28 msec) and Hercules Monochrome Card	$1,899.00
Model K401-v with 4 MB on-board DRAM, 40 MB Hard Drive (28 msec) and VGA Card	$1,999.00
14" VGA Monitor, 0.31 mm, Gold Star 1450 Plus with K401-v or K402-v (Regular $549.00)	$ 449.00
14" Monochrome TTL Monitor, Amber, Gold Star 1401A with K401-m or K402-m (Regular $199.95)	$ 149.00
VGA Monochrome Monitor, white, Tandy VGM-100 with K401-v or K402-v (Regular $199.95)	$ 165.00

OS9/680x0 SOFTWARE

SCULPTOR - Development System (68000) from	$2,500.00
Quick Ed - Editor and Text Formatter	$ 275.00
FlexiLint - A must for C programmers	$ 495.00
Caching - High speed disk caching (demo available)	$ 300.00
IMP - Intelligent Make Program	$ 250.00
Disassembler - 3 pass	$ 250.00
Windows - C source code & library	$ 250.00
Profile - Tune User State Programs	$ 270.00
PAN Utilities - C source and library	$ 250.00

delmar co

P.O. BOX 78 • MIDDLETOWN SHOPPING CENTER • MIDDLETOWN, DE 19709 • 302/378-2555