



microware[®]

Covering the Full Spectrum of 6809 and 68000 Applications of OS-9

PIPELINES

NEW C-SOURCE-LEVEL DEBUGGER...Page 2

MICROWARE OPENS WESTERN REGIONAL OFFICE...Page 3

ON THE C SIDE...Pages 4 & 5

NEW DEVELOPMENT PAK...Page 7

MICROWARE ON-LINE
COMPUSERVE SUPPORT...Page 8

PIPELINES

Fall/Winter 1987
Volume 2 Number 4

Pipelines is published quarterly by:
Microware Systems Corporation
1900 N.W. 114th Street
Des Moines, IA 50322
(515)224-1929

Editor:
David F. Davis

Contributing Editors:
Andy Ball Dave West James Jones

Photography Art Director
David F. Davis George Edwards

A NOTE TO OUR LOYAL READERS

As with every issue of Pipelines, we would like to solicit your comments, suggestions and thoughts. Also, we invite you to send for publication any interesting OS-9 application articles or useful utility programs. Address submissions to the Editor of Pipelines, c/o Microware Systems Corporation. Please include your name, address and telephone number.**

OS-9 EDUCATION AND TRAINING SEMINARS

On a regular basis, Microware Systems conducts a two-day seminar for hardware manufacturers, OEM's, end-users or others who require hands-on training on the design, implementation and operation of OS-9/68000. Microware hosts all seminars on site in our Des Moines, Iowa facilities where you will have access to our development systems and software engineers. Training fees include all courseware and copies of related documentation. Please contact Microware regarding seminar schedules and fees.**

EXCITING NEW PRODUCTS FROM MICROWARE

Microware announces three new products for the OS-9/68000 Operating System: Source Level Debugger, TCP/IP Ethernet Support and System Protection Utilities. All three products are scheduled for a December release.

C SOURCE-LEVEL DEBUGGER

Presenting a powerful C Language programming tool--and one that's easy to use! The debugger contains features commonly found in source-level debuggers, as well as a number of unique and innovative extensions designed for the OS-9 programming environment.

The C source-level debugger includes a full-featured C expression interpreter that permits the programmer to call a function with parameters in the program being debugged. The result can then be printed, a break point set and programmer can then step through the function. This feature allows the testing of functions separately in the program being debugged.

The expression interpreter also supports the new Microware C Compiler data types and operations (enumerators, bit fields, structure assignments, functions that return structures and structure parameters). This provides programmer with total compatibility between the source level debugger and the C Compiler.

Another unique feature is sub-line expression stepping. The debugger prints a character pointing to the expression in the source line that is to be executed next. If there is more than one statement in a line, then the debugger will step across the line to the next statement. This allows the programmer to watch each expression as it is executed in a complex C statement.

These features--in addition to the more fundamental characteristics such as on-line help, tracing, break points, watch expressions and variable display/change facilities--make this new Microware C source-level debugger a powerful programming tool. The debugger not only speeds program development, but makes C language programming a tool that can be utilized by neophyte programmers.

ETHERNET SUPPORT

Microware's new Ethernet Support package utilizes the popular TCP/IP protocol to facilitate communications between OS-9/68000-based systems as well as other operating system environments.

This initial release of the Ethernet protocol package is designed to support the Communication Machinery Corporation (CMC) ENP-10 Ethernet controller. Subsequent releases to support additional hardware, including the Motorola MVME 330-A-K controller, are under development.

Microware's TCP/IP implementation conforms to the United States Department of



A sample session of
source-level debugging.



MICROWARE OPENS WESTERN REGIONAL OFFICE

On September 1, 1987, Microware opened a Western Regional Office in Santa Clara, California. The opening of this facility, located at 4401 Great America Parkway, Santa Clara, CA 95086 (408-980-0201), reflects Microware's growth and commitment to comprehensive customer service.

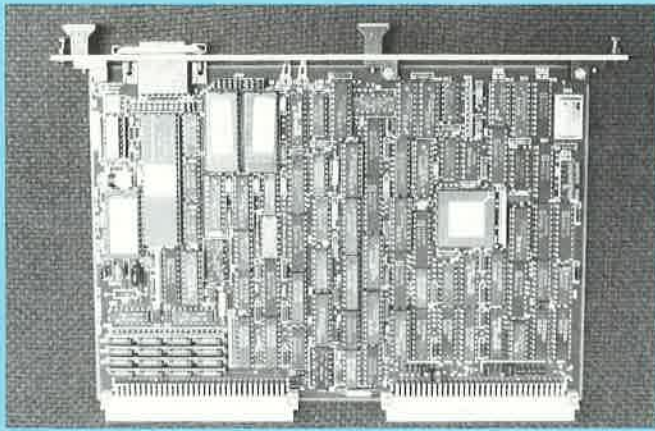


Andrew Crane
Western Regional Sales Manager

Heading up Microware's new office is Andrew Crane. Andrew has previously held positions as Technical Sales Representative and Domestic Sales Manager at our Des Moines, Iowa headquarters. Working with Andrew as Office Manager is Pamela Johnstone. Two technical sales representatives and a software engineer will soon be added.

The new Western Regional Office has responsibility for all marketing activities for customers in the states of Arizona, California, Nevada, Oregon, Washington, Utah and Hawaii. Educational seminars, customer technical or technical services for OEMs or end-users may be arranged through this office.

A variety of OS-9-based systems are available on-site for demonstration, research and development or educational purposes. Anyone in the Western Region who has questions, or needs additional information, should contact Andrew.**



**The CMC ENP-10
Ethernet controller.**

Defense ArpaNet standards. This standard incorporates FTP (a popular file transfer protocol) and Telnet (a virtual terminal interface). FTP allows the user to transfer files and data between Ethernet nodes. Telnet allows the user to login on remote systems over the Ethernet-TCP/IP network.

Microware selected the Ethernet protocol because of its rapid growth in popularity as a networking standard. The Ethernet package allows the user to gateway from the OS-9 Operating System to UNIX, VMS and other operating system environments. This communications path provides connectivity for distributed program development and application systems.

The Microware Ethernet support package includes a new OS-9 file manager that supports a BSD 4.2 type socket interface and a new CMC ENP-10 device driver.

SYSTEM PROTECTION UNIT MODULE

Microware's new memory protection module--System Protection Unit (SPU)--provides software memory protection for a Memory Management Unit and other custom devices.

The SPU module uses a system-wide permission mechanism to limit RAM access for user state tasks. Hardware devices supported by Microware's SPU module include the MC68451, MC68851 and custom devices. The SPU software is delivered as a modular addition to OS-9 in source code form and is divided by functions into separate routines that can be modified to support different memory protection devices.



The Motorola 68851 PMMU.

SPU memory protection provides both a secure environment for development system work and a significant aid in program debugging. The SPU module provides program run time protection, protection against wild pointers and detection problems before CPU crash.

The SPU software comes complete with a new permission memory map utility--MAPS. This utility graphically displays the memory address space using OS-9 Termcap facility. MAPS displays information about process registers and the areas of RAM that each process can read and write.

Contact Microware or an Authorized Microware Distributor today for packaging and pricing information about any of these new products.**

ON THE C SIDE

Among the differences between Microware's OS-9/6809 shell and OS-9/68000 shell is the addition of environment variables to control the behavior of the shell and of other programs that establish the convention of looking for particular environment variables. How is this done? Perusal of the **F\$Fork** and **F\$Chain** system call manual pages shows no modifications to pass an environment. The answer, like that to the question of how C programs can be handed the **argv[]** and **argc** they expect, lies in the parameter string and in the setup code in **cstart.r**.

Just as the C startup code parses the parameter string into **argv[]** and determines the associated argument count, it now also creates an array of (name, value) pairs for environment variables from the same parameter string. To maintain compatibility with early versions of the C startup code, the arguments appear first, with the CR delimiter they have always had. They are in turn followed by the names and values of the environment variables. A C program's **main()** function can now read:

```
main(argc, argv, envp)
int      argc;
char     *argv[];
char     *envp[];
{
    /* et cetera... */
}
```

giving one access to the environment that way, or one can use the **getenv()** function, passing an alleged environment variable name and being returned either a NULL pointer or a pointer to the string which is the value of the environment variable requested.

Do you as a programmer have to put together the parameter string that **cstart** parses all by yourself? No. There is a C library function, **os9exec()** that will do it for you, and also spawn a process. Quoting the manual page:

```
os9exec(procfunc, modname, argv, envp,
stacksize, priority, pathcnt)
int      (*procfunc)();
char     *modname;
char     *argv[];
char     *envp[];
unsigned stacksize;
short    priority, pathcnt;
```

Unix programmers will note a resemblance between **os9exec()** and the Unix **execve()** function. For them, a brief listing of the differences:

- Unix **exec()** most resembles the OS-9 **F\$Chain** system call. One can pass **chain** or **chainc** as **procfunc** to achieve that effect, but one can also pass **os9fork** or **os9forkc** so that **os9exec()** can spawn processes as well.
- The **stacksize**, **priority**, and **pathcnt** arguments modify

aspects of processes that would be modified in other ways under the Unix environment, e.g. by directly closing paths or setting one's priority. Since **os9exec()** can spawn as well as chain processes, these arguments are provided to give the parent process control over the child.

procfunc is a pointer to a function that either creates a process or chains the current process to a new primary module. Typically this will be one of **os9fork**, **os9forkc**, **chain**, or **chainc**.

modname points to the name of the primary module of the new process, or, if one is chaining, the new primary module for the current process.

argv should point to an array of pointers to strings that will be the arguments for the new process or chained old process. By convention, **argv[0]** should be the name of the primary module, and a NULL pointer should terminate the array.

envp should point to an array of pointers to strings that represent the environment variables to be seen by the process. These are a collection of strings of the form "name=value", terminated by a NULL pointer just as **argv** is.

stacksize indicates the additional memory (over and above that asked for in the module header) to be given to the new or chained process. In particular, if **stacksize** is zero, the default as indicated in the module header will be used.

priority indicates the priority to be given to the new or chained process. If priority is zero, the new process will inherit the priority of the old.

pathcnt should be specified only if you used **os9forkc** or **chainc** as the **procfunc** argument, and indicates the number of I/O paths to be inherited by the new process.

os9exec() returns the value returned by its **procfunc** argument when it is invoked with values derived from the other arguments to **os9exec()**. (Of course, **chain** or **chainc** will return only if they fail.) This will either be -1, indicating an error, or a process ID for the parent's later use.

Microware's OS-9/68000 shell does most environment manipulation, via its built-in commands "**setenv**" and "**unsetenv**." As an example of their use, here's a portion of a shell **.login** file:

```
setenv PROMPT "@Ack. Thtpf. "
setenv TERM kt7
setenv PATH ../../NET
setenv EDITOR umacs
setenv _sh 0
setenv TZ CST
```

PROMPT, PATH, and **_sh** are used by the shell; TZ is used by some C library functions that manipulate times; TERM is

ON THE C SIDE

used by the termcap library. EDITOR indicates one's preference in text editors to the Microware mailer.

Probably the first time you as a programmer will run into the environment will resemble this: you write or port a program which lets you escape to the shell, and one day while running it you use that feature. Suddenly your prompt is a drab "\$", and you can't get to a directory that contains some highly useful programs any more unless you use absolute pathnames. You've been bitten by a program that blithely does an `os9fork()`, which does not propagate the environment your `.login` file (and you, if you've modified it) has established.

Such programs should be converted to use `os9exec()` instead, so that the environment makes it through to the child process. Sharp-eyed readers will be worried at this point, since we've not mentioned any way, other than via arguments to `main()`, that one can access one's environment. Fortunately, including the declaration

```
extern char *environ[];
```

will give you access to your environment for passage on through `os9exec()`.

The following example is taken from a modified version of the public domain QED editor in the OS-9 User's Group Library, and shows the relatively small changes needed to convert an OS-9/6809 program to OS-9/68000 so that the environment is passed on to a child process. (Here "scan" is a character pointer that points at the user's input line, and the function is designed to return NULL if all is well, or a pointer to an error message string if problems arise.) Rather than require the editor to scan the line and create an argv array, it passes the buck on to the shell.

(NOTE: A more parsimonious version would prepend "ex " to the string, so that the shell will only exist as long as it takes to parse the string and in turn `os9exec()` the module we really wanted. Should you wish to try your hand, remember that there will be a special case to check for, i.e. that in which scan points to a string containing only whitespace. Then one should simply spawn an interactive shell, so that "ex " should not be prepended.

```
#define CANT_FORK "can't fork"
#define BAD_EXIT "abnormal exit"
int      os9exec();  os9fork(),  wait();
#ifdef OSK
extern char **environ; /* pass on the environment we have now */
char      *argv[] = {
    "shell", /* the name of the program to be invoked. a place */
    NULL,   /* holder, to be overwritten with a pointer to */
    NULL};  /* the input line the shell will parse for us */
#endif
char *ShellCmd()
{
    int      Sonny, ErrCode; /* child process, and its exit code */
#ifdef OSK
    argv[1] = scan;
#else
    char      *ShParm;
    ShParm = scan;
#endif
    /* Skip the command line, lest it start with "l" or "p" and thus confuse */
    /* the main command interpreter. */
    while (*scan != '\n') scan++;
#ifdef OSK
    Sonny = os9exec(os9fork, argv[0], argv, environ, 0, 0);
#else
    Sonny = os9fork("shell", scan + 1 - ShParm, ShParm, 1, 1, 0);
#endif
    if (Sonny == -1) return(CANT_FORK);
    while (Sonny != wait(&ErrCode));
    if (!DoPrompt) fputs("!\n", stderr); /* let the user know we're done! */
    return(ErrCode == 0 ? NULL : BAD_EXIT);
}
```

NEW EMPLOYEES



**Front Row (l-r): Josef Laake, Denny Bridal, Lee Glenn, Steve Johnson and Rik Yeates.
Back Row: Lori Lockard, Sandy Lowe, Sandee Cook, Terri Rieger and Kristin Doane.**

Microware is on the move! Here are the latest additions to our growing staff.

Josef Laake, Software Engineer, is involved with adapting OS-9 to new hardware systems. Previously, Josef worked for Cross Technologies as a small systems designer.

Denny Bridal, Technical Sales Representative, formerly worked as a sales representative for Cadalyst, Inc. marketing CAD systems. Denny has a B.S. in economics from Simpson College in Indianola, Iowa.

Lee Glenn is one of the new voices on the Microware Hotline. He has a degree in Computer Science from Simpson College and has been using OS-9 on his home system since 1983.

Steve Johnson, Technical Sales Representative, comes to Microware from Mizar where he was a product manager. Steve has an E.E. degree from the University of Minnesota and is currently working on his M.A. in Business Administration.

Ric Yeates is also a new Microware Hotline voice. Ric is completing a B.S. degree in Computer Science from Iowa State University. Ric says he enjoys working with Microware customers and systems level software.

Lori Lockard is a Production Assistant in Microware's Production Department. She is a 1984 graduate of Roosevelt High School and has attended Grandview College in Des Moines for 2 years studying Computer Science.

Sandy Lowe, Accounting Assistant, is in charge of all accounts receivable for Microware. Sandy has worked as an accounting clerk for several companies in the Des Moines area, and enjoys sewing, needle point and golf in her spare time.

Sandee Cook recently joined Microware as an Accounting Assistant and Secretary in charge of Accounts Payable and Personnel. Sandee previously worked as a purchasing agent for a bottling company in the Des Moines area.

Terri Rieger, Receptionist, is the new voice of Microware. Terri operates Microware's PBX system. She previously worked as a secretary/bookkeeper for Calhoun-Burns & Associates.

Kristin Doane is the Marketing Department's new Account Administrator. Kristin will be responsible for maintaining the OEM data base, accurate record reports and departmental correspondence.

(Not Pictured)

Kathie Flood has joined the Documentation Department as a Technical Writer. She has a B.S. in Math/Computer Science from Central University of Iowa, and is working on her M.A. in Journalism at Drake.

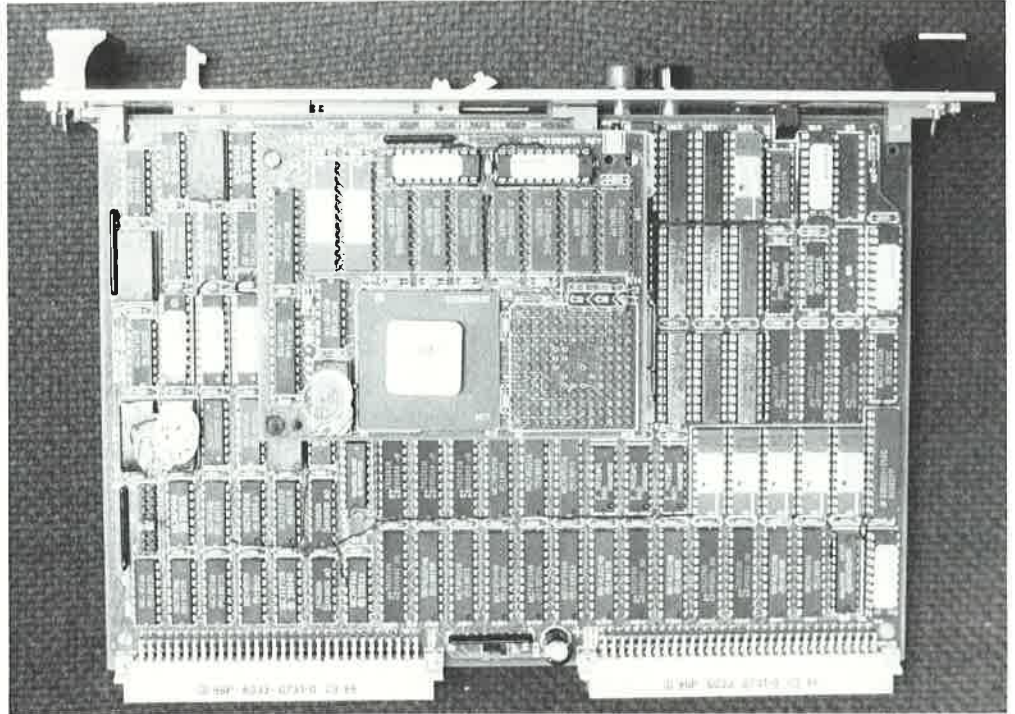
Pamela Johnstone is the Office Manager for Microware's new Western Regional Office in Santa Clara, CA. She is responsible for all office functions and related activities. Pamela previously worked for First National Mortgage in the purchasing and personnel departments.**

NEW DEVELOPMENT PAK FROM MICROWARE

MVME 130/131 NOW AVAILABLE.

A Development Pak for Motorola VME 130DOF, 131 or 131XT CPUs is now available from Microware. The MVME 130/131 is a popular 68020 CPU designed for industrial applications. The new Development Pak includes device driver support for the MVME 050 System Controller. This controller provides two additional serial ports, a parallel port and a real-time clock. Floppy and Hard disk support is provided for a MVME 320 Disk Controller. As with all Microware/Motorola Development Paks, this version of OS-9 is configured to support up to 16 megabytes of RAM.

Contact Microware or an authorized Microware distributor for complete Development Pak pricing and distribution information.**



The MVME 131XT 32-bit monoboard microcomputer

HELP!

We Need to Clean-Up Our Mailing List.

Since the publication of our charter issue of Pipelines in September, 1981, the mailing list has grown to dwarf the Tokyo Telephone Directory. We are now faced with the decision of either charging for Pipelines to cover the cost of postage, or reviewing our mailing list. Andy Ball, our V.P. of Marketing, has suggested the latter. To assure that you continue to receive Pipelines (without

interruption and without cost) **YOU MUST FILL OUT AND RETURN THE INFORMATION COUPON BELOW.** Please send the completed form to: Editor, Pipelines c/o Microware Systems Corporation, 1900 N.W. 114th Street, Des Moines, Iowa 50322. If there is someone you know who would also like to be on our mailing list, please include their name and address, too.

Name _____ Title _____

Company _____ Phone () _____

Address/P.O.Box _____

City/State/Zip _____

MICROWARE ON-LINE SUPPORT COMES TO COMPUSERVE



Now you can get around-the-clock, on-line technical support from Microware on the CompuServe information Service. CompuServe, the nation's largest information service, can be accessed by a personal computer, a phone modem, communications software and a CompuServe account, user's guide and an introductory credit of \$25.00, are available at

many computer stores, bookstores, or from CompuServe

Once on-line, simply type "GO MSC" or "GO MICROWARE" at any prompt to enter Microware's display area. When you enter the display area, you will be presented with the following menu:

MICROWARE ONLINE SUPPORT

- 1 Using This Service
- 2 What's New at Microware?
- 3 Sales and Marketing Info.
- 4 OS-9/68000 User Notes
- 5 OS-9/68000 PDRs
- 6 OS-9/6809 User Notes
- 7 More Software Support
- 8 Contacting Microware
- 9 Send a Message to Microware

- an overview of available topics
- new products, official announcements
- product descriptions and distribution information
- 68000 "how-to's", release notes, tutorials
- software problems for OS-9 Version 2.1
- 6809 "how-to", release notes, tutorials
- purchasing additional software support
- address/phone for Microware and distributors
- ask questions, make suggestions

Enter choice!

The entire display area is completely menu-driven allowing even beginning users to quickly familiarize themselves with the available information. Expert CompuServe users can consult their User's Guide for even faster access methods. Where applicable, topics are categorized by release version number and product description to eliminate unnecessary reading. New topics are constantly being added, so check often for subjects of interest to you.

For a list of local CompuServe access telephone numbers or to order a CompuServe subscription kit, simply call CompuServe between 8:00 AM and 10:00 PM, Monday through Friday, Saturday between 12:00 noon and 5:00 PM. In the United States, call 1-800-848-8199. In Ohio, or outside the United States, call (614) 457-0802.**



MICROWARE SYSTEMS CORPORATION
1900 NW 114th Street
Des Moines, Iowa 50322

Bulk Rate
U.S. Postage
Paid
Des Moines, IA
Permit #2864