SUN MICROSYSTEMS COMPUTER CORPORATION

SunOpsis

The Solaris 2.0 Migration Support Centre Newsletter

1-800-363-6200 VOLUME 1, NUMBER 1, AUGUST 1992

Editorial

Collette Moquette Ricks, Solaris Product Marketing Manager, NAAFO
Collette.Ricks@EBay.Sun.Com

Welcome to the premier issue of SunOpsis The Operation Commitment monthly newsletter.

Operation Commitment (OPCOM) is a program created by Sun Microsystems to assist you, our key customers in the migration to Solaris 2.0.

In order to give you a head start, OPCOM customers had access to an early version of Solaris 2.0 prior to First Customer Shipment. OPCOM customers will continue to have early access to Solaris Releases as well as access to a Porting Support Line that you can call when you have technical questions during your migration process.

This newsletter is another resource that OPCOM customers will be able to rely on. Each month several feature articles of a technical nature will be presented along with some regular columns. One such column is our "Solaris 2.0 Hints and Tips" section which includes ideas and suggestions gleaned from the numerous calls we receive.

Several articles appearing this month and in the coming months include source code and examples. These are available from our ftp server and also by electronic mail.

We hope that you will find the OPCOM efforts helpful and continue to let us know how we can further assist your efforts. We would like to thank you for participating in the Operation Commitment Solaris 2.0 migration program.

Calling Solaris 2.0 Support Dispatch

Lisa Thompson / Deborah Hummel

We can be reached at 1-800-363-6200 or email:
solaris2@sun.com

Your call will be answered by one of our dispatchers:

Deborah Hummel
Lisa Thompson

Information obtained will then be used to open up a service call on your behalf and will be forwarded to the appropriate engineer. The engineer will then respond within the standard 1 hour call back time. Email response is much the same, however an email will be returned to you by the Dispatcher letting you know that the call has been logged and giving you the Call # for your information.

INDEX

OPCOM REGISTRATION

Lisa Thompson / Deborah Hummel

How is support determined and how do I get there from here?

Customers who qualify for support services is determined by Solaris 2.0 Migration Managers.Your SUN Sales Representative will correspond with management staff and forward the necessary information required for registration.This registration is then sent to OPCOM for verification purposes when customers call for support.

When there are revisions, deletions or additions to be made to registrations which are currently in the data base you must contact the dispatchers at OPCOM or send email to:

solaris2-register@sun.com

with this information to update your record. Please keep in mind this includes stand-ins for vacation coverage.

Introducing the Opcom FTP Server

Georg.Nikodym@Canada.Sun.Com

At OPCOM, we maintain an anonymous FTP server which is a repository for many Solaris programs and suites. These include ported program binaries and source, documentation, driver porting information and much more.

The host name of the server is `opcom.canada` (IP 129.155.8.88) and currently has four major areas. These are `binaries`, `docs`, `drivers`, and `tars`.

In the `binaries` section you'll find executable programs and related man pages for software that we and others have ported to Solaris 2.0. These include `a2ps`, `ftptool`, `gnudiff`, `less`, `olvwm`, `patch`, `tcsh`, `workman`, and `xvnews`.

The `docs` section contains documents on networking, the Internet RFCs, on SCSI, Solaris migration, and various white papers.

If drivers are your interest, then look at the `drivers` section which is home to a fair amount of interesting items related to driver writing under Solaris 2.0, including a skeleton SCSI driver, and a loadable system call module.

The `tars` section, as its name implies, contains compressed tars of a software that has been either ported or developed here. Of particular interest is Ron Winacott's batch scheduling class, and an alpha port of X11r5.

If you wish to place things on this server to share with the world, you can upload to the `tmp` area.

Notes about source code on this server:

- Source is supplied as is. As of this writing, many of the PD packages are the most recent available, but this is not guaranteed to remain the case.
- Porting information, in general, has been made available to original authors.
- No source license required. Licensed folk may contact Dean.Kemp@Canada.Sun.Com to obtain source not on this server.

If you have any problems, questions, or wish to have your own items made available on this server, send mail to Georg.Nikodym@Canada.Sun.Com. If you cannot access this FTP server, send email to ftp@opcom.Canada.Sun.Com with the subject 'help'

Setting Up an Anonymous FTP Server On Solaris 2.0

Gene.Saunders@West.Sun.Com

Setting up an anonymous ftp server is not always as straightforward as it may seem. Furthermore, the requirements have changed from Solaris 1.0.1, making the procedure even less clear. This article provides a 'step-by-step' method for creating a working ftp server on Solaris 2.0.

1. Choose the directory to serve as the anonymous FTP root directory. For example, /export/ftp.
2. i) Login as root `(or use su to become root)`

   ii) start `/usr/openwin/bin/openwin` if it's not already running

   iii) start `/usr/bin/admintool` once the window system is up.

   a) select Database Manager.

   b) select Naming Service: None.

   c) select Passwd and press "Load".

   d) choose "`Edit->Add Entry`", and make an entry for username "`ftp`" using the home you previously selected and a `uid/gid` pair of your choice.For our example, select uid 11, gid 11 with home directory field set to /export/ftp and the shell field set to the string "*NOSHELL*" (for added security). For the Password Status field, use "no password --- "

e) press "Add" to add the entry to your passwd file.

iv) quit the admintool

3. Create your directory tree. Our example is fairly typical, with a /pub directory for downloads and a /tmp for uploads. To make the directory structure, execute (as root) the following commands:

```
# mkdir /export/ftp
# cd /export/ftp
# mkdir bin dev etc pub tmp usr usr/lib
# chmod 555 . bin dev etc pub usr usr/lib
# chmod 777 tmp
# chown bin . bin dev etc pub usr usr/lib
# chown ftp tmp
```

This will enable anonymous ftp users to write files only to /tmp, thereby eliminating a security risk from having write access to other areas.

4. Create the entries in /export/ftp/dev:

```
# cd /export/ftp/dev
# (cd /dev; tar cvhf - tcp zero) | tar xvf -
# chmod 644 tcp zero
```

5. Retrieve the necessary files from the Solaris distribution:

```
# cd /export/ftp/usr/lib
# for i in ld libc libdl libintl libw; do
> cp /usr/lib/$i.so.1 .
> done
# chmod 444 *
# chmod 111 ld.so.1
# cp /etc/netconfig /export/ftp/etc
# chmod 444 /export/ftp/etc/netconfig
# cp /bin/ls /export/ftp/bin
# chmod 111 /export/ftp/bin/ls
```

This completes a minimal setup. In this configuration, the ftp DIR command will show numeric user id's and group id's. If you prefer the ftp users to see user and group names instead, do the following additional steps:

```
    # cp /etc/group /export/ftp/etc/group
    # cat > /export/ftp/etc/passwd
root:*:0:1:0000-Admin(0000):/:*NOSHELL*
bin:*:2:2:0000-Admin(0000):/bin:*NOSHELL*
ftp:*:11:11:The FTP account:/:*NOSHELL*
^D
    # cd /export/ftp
    # cp /usr/lib/nswfiles.so usr/lib
    # chmod 444 etc/group etc/passwd
    # chmod 444 usr/lib/nswfiles.so
```

6. Test your setup, by trying to FTP to yourself and requesting a directory listing, and then putting a file there:

```
% ftp gesundheit
Connected to gesundheit.
220 gesundheit FTP server (UNIX(r) System V
Release 4.0) ready.
Name (gesundheit:saunders): anonymous
331 Guest login ok, send ident as password.
Password:
230 Guest login ok, access restrictions
apply.
ftp> dir
200 PORT command successful.
150 ASCII data connection for /bin/ls
(127.0.0.1,2834) (0 bytes).
total 12
drwxr-xr-x 2 bin other 512 Aug 6 22:13 bin
drwxr-xr-x 2 bin other 512 Aug 6 21:47 dev
drwxr-xr-x 2 bin other 512 Aug 6 23:02 etc
drwxr-xr-x 2 bin other 512 Aug 6 22:45 pub
drwxrwxrwx 2 ftp other 512 Aug 6 22:45 tmp
drwxr-xr-x 3 bin other 512 Aug 6 22:19 usr
226 ASCII Transfer complete.
364 bytes received in 0.18 seconds (2 Kbytes/
s)
ftp> bye
221 Goodbye.
```

If you were able to get similar results to the example above, then your FTP server is functional now. A shell script to set up anonymous ftp is available from the OPCOM FTP Server.

The Device Driver Interface/Driver Kernel Interface

Srinath.Venkat@Corp.Sun.Com

In System V Release 4.0 (SVR4), the interface between device drivers and the rest of the Unix kernel has been standardized and is now documented in section 9 of the Solaris 2.0 Reference Manual. This section documents driver entry points, driver callable functions and kernel data structures used by the device drivers.

These interfaces, known collectively as the Device Driver Interface/Driver-Kernel Interface (DDI/DKI) are divided into three general groups:

- The DKI-only interfaces that are specific SVR4. Note that these interfaces may not be supported in future releases of System V.

- The DDI-only interfaces, which are architecture specific.

- The remaining interfaces that are generic and are included in both DDI and DKI.

Sun's implementation of the DDI/DKI was designed to provide binary compatibility for third party device drivers across currently supported hardware platforms and minor releases of the operating system. However, unforeseen technical issues may force changes to the binary interface of the DDI/DKI.

Furthermore, future releases may contain additions to the DDI/DKI to support future platforms. At that time device drivers wishing to operate across the new set of supported platforms may require some changes.

SunDDI is the Sun architecture-specific(DDI) part of the Solaris 2.0 DDI/DKI. The Solaris 2.0 DDI/DKI, like its AT&T counterpart, is intended to standardize and document all interfaces between device drivers and the Unix kernel. In addition, the Solaris 2.0 DDI/DKI is designed to allow source compatibility for drivers on any Solaris 2.0-based machine, regardless of the specific machine architecture or processor. As well, it will provide binary compatibility for drivers running on any SunOS 5.0-based machine which has the same processor, regardless of the specific machine architecture.

The Solaris 2.0 DDI/DKI will allow platform-independent device drivers to be written for Solaris 2.0 -based machines. These *shrink-wrapped* drivers will allow third-party hardware and software to be integrated into Solaris 2.0 -based machines of the same arch much as they are on IBM-compatible personal computers today

Installing A NIS+ Root Server On A 4.1.X System

Enrico.Giuditti@Canada.Sun.Com

The following article will outline the steps required to set up a NIS+ Root Server on a 4.1.x system. (Note you must be a root user to execute these commands.

1) Mount the Solaris 2.0 CD either locally or remotely from another machine. To mount it locally:

   a) find a mount point, if you do not have one, create it with the command: `mkdir /cdrom`

   b) mount the CD: `mount -F hsfs -r /dev/dsk/c0t6d0s0 /cdrom`

Now extract the necessary files. Note: You will want to make a backup copy of these files since they will be overwritten by the tar extraction: /usr/bin/keylogin, /usr/bin/keylogout, /usr/etc/keyserve

   c) go to the root directory: `cd /`

   d) tar the file NISPLUS.TAR: `tar xvf /cdrom/NISPLUS.TAR`

(Note: It is important to be in the root directory because all the tar files have absolute pathnames)

2) Create the directory /var/nis: `mkdir /var/nis`

3) Be sure your domainname is defined: `cat /etc/defaultdomain` This should return a valid domain name eg: foo.bar.sun.com. If you don't have a valid domainname, set it using the domainname command:

```
# domainname foo.bar.sun.com
# domainname >/etc/defaultdomain
```

4) Kill the keyserv process currently running so the latest keyserv can be executed:

```
# ps -x | grep keyserv
# kill <pid_for_keyserv>
```

5) Remove the file /etc/.rootkey: `rm -f /etc/.rootkey`

6) Start the process /usr/etc/keyserv: `/usr/etc/keyserv`

7) Execute `/usr/sbin/nisinit -r` This will create the file /var/nis/<systemname>/root.object so the root server can identify and initialize itself upon bootup.

8) Execute `/usr/sbin/rpc.nisd -r -S 0`. Rpc.nisd is the daemon that provides ONC-RPC service which implements the NIS+ service. This will also give you security level 0 which is needed because no credential table exists yet.

9) Now setup an environment variable "NIS_GROUP" which is the default group identifier for the group owner. This is required because NIS+ objects are assigned an owner and a group when they are created. By setting this variable now will save you having to change it later for every object.

   eg: if the admin group was named *admins* and domain name *foo.bar.sun.com* :

```
setenv NIS_GROUP admins.foo.bar.sun.com.
```

   or

```
NIS_GROUP=admins.foo.bar.sun.com
export NIS_GROUP
```

10) Execute `/usr/lib/nis/nissetup` or `/usr/lib/nis/nissetup -Y` for NIS compatibility mode. Nissetup will create the default tables on the server (eg hosts passwd rpc) to allow clients to store system administration information. If you choose YP compatibility nissetup assigns read permissions to the *nobody* category in all the objects that it creates, allowing NIS clients, who are unauthenticated, to get information from the NIS+ tables. (Note: If NIS compatibility mode is used here it must be included when starting the rpc.nisd daemon.)

11) Execute `/usr/bin/nisaddcred des` Nisaddcred adds DES security credentials for NIS+ clients. It stores these credentials in the table *cred.org_dir* in the default NIS+ domain. It will ask you for the root passwd and for verification.

12) Execute `/usr/bin/nisgrpadm -c group-name` This will create the group *group-name* owned by the NIS_-GROUP variable set in step 9.

    /usr/bin/nisgrpadm -c admins.foo.bar.sun.com.

    Use the -a option to add members to the group.

13) Execute `/usr/etc/nis/nisupdkeys <domain>`

14) Execute: `/usr/etc/nis/nisupdkeys org_dir.<domain>`

15) Execute `/usr/etc/nis/nisupdkeys groups_-dir.<domain>` Nisupdkeys will update the public keys in an NIS+ directory object. The domain must be fully qualified.

16) Kill the rpc.nisd process currently running so that a new rpc.nisd with DES security can be executed.

```
# ps -x | grep rpc.nisd
# kill <rpc.nisd_pid>
# /usr/etc/rpc.nisd -r
```

    (for YP compatibility mode execute /usr/etc/rpc.nisd -r -Y)

17) Execute `/usr/bin/nisaddent -r -f <input file either NIS or file based> <NIS+ table>` This will take an NIS or files entry and convert it to NIS+ table. This must be done for each table eg: hosts passwd....

You have now setup a 4.1.x NIS+ root server. The last phase is to edit your /etc/rc.local file to start the NIS+ server whenever the system is booted. Add these lines after the invocation of NIS.

```
# Start NIS+ if the service is installed and
   initialized.
if [ -d /var/nis/$HOST -a -x /usr/etc/rpc.nisd
   ]; then
   if [ -f /var/nis/$HOST/root.object ]; then
      rpc.nisd -r;echo -n 'NIS+ (root server)'
   else
      rpc.nisd; echo -n 'NIS+'
   fi
fi
```

(Note: If you used -Y option (YP compatibility mode) in step 16 replace the above `rpc.nisd -r` to `rpc.nisd -r -Y`.

Porting Experience: Epoch and Perl

Gerry.Singleton@Canada.Sun.Com

When dealing with large programs such as perl or epoch, you can make the effort a bit easier by paying particular attention to a few details that you may have ignored in the past. In porting both perl and epoch to Solaris 2.0 we have discovered a few techniques that seem to apply in these cases.

First, you should carefully consider the order of all the libraries that are linked into your program. Often the stock configuration selections (whether generated by configuration routines or supplied from pre-configured files) may not be the best choice.

For example, perl's configuration routine includes -lmalloc along with its choices for libraries to include. If you accept this default the result will be a very slow perl. An assumption is made that the system malloc is mutually exclusive with perl's malloc. Since there are no rules to assist you in the determination you *must* discover the best order empirically, based upon your programming experience.

Second, use compiler options that provide the most general behavior. For example, with the SUNPro compiler use the default -Xt option whenever possible. With this option the compiler will generate ANSI warnings that are generally harmless and it will also compile the program using a compilation environment very similar to the original.

Third, you should use a conversion filter to prepare your program for ANSI compliance. There are a number of tools that provide assistance. I've found `proto` and `unproto` to provide quite reasonable results.

Fourth, `lint` the source. This step is often ignored because the large amount of output is difficult to digest. "Bite the bullet" and look at it now. You can save yourself hours of debugging time. Lint is an excellent tool for catching small incompatibilities that can haunt you in your quest for a successful port.

SOLARIS 2.0 HINTS AND TIPS

The entries in this section are a series of interesting items discovered by users of Solaris 2.0

We have selected just some of them for this months column. For a complete listing please check the 'hints-and-tips' file available on the OPCOM FTP server. To submit an entry, simply e-mail it to 'solaris2@Sun.Com' with an appropriate subject line.

Augmenting The Fcode in Your SBUS Prom

Gil.Hauer@Canada.Sun.Com

Solaris 2.0 provides a mechanism that allows non self-identifying hardware to be configured correctly in the kernel. This is called a "hardware configuration file". It typically lives in /kernel/drv and is usually named `xx.conf` where `xx` is the name of the driver that controls the device.

Although it is generally used to aid the configuration of devices, it can be used to augment the information contained in the FCODE Prom on the SBus card.

To add properties to the device you first have to provide an exact match of the *name* property, and the first *reg* property. Properties following these definitions in the configuration file will be copied into the kernels dev_info tree.

Communicating Drivers

Gil.Hauer@Canada.Sun.Com

Do your drivers need to share information?

The first step is to create a property that is owned by the parent device. The property can then be read by all of the children of the parent (i.e. the siblings of the device driver).

To do this you will need the dev_info pointer of the parent. This can be accessed by `ddi_get_parent()` DDI/DKI call. Once you have the parents device information structure you can create a property using the `ddi_prop_create()` DDI/DKI call.

The siblings of the driver can then read that property using one of the `ddi_prop_op()` DDI/DKI calls.

Installing a Terminal With Solaris 2.0

David.Guerra@Canada.Sun.Com

The following is a procedure for installing a terminal on serial port A under Solaris 2.0.

1. Make a proper null modem cable (Pins 2,3,7 are not sufficient). *Refer to your Sparcstation installation manual for a description of the correct pinouts.*

2. Initialize a TTY port monitor, 'ttymon' using the command

```
# sacadm -a -p ttymon0 -t ttymon -c
   /usr/lib/saf/ttymon -v `ttyadm -V`
```

3. Display the port status with the following command

```
# sacadm -l -p ttymon0
```

4. You must now initialize the serial line with the '`pmadm`' low level service command.

```
# pmadm -a -p ttymon0 -s a -i root -v `ttyadm
   -V` -f u -y "ttya" -m "`ttyadm -i
   'TERMINAL DISABLED.' -d /dev/term/a
   -l conttyH -m ldterm,ttcompat -p
   'TERM login: ' -s /usr/bin/login`"
```

NOTE: This command must be typed on a single command line. Also, extreme care must be taken to use the correct quotation marks.

This should result in the display of a login prompt, such as: TERM login:

5. You can disable the ability to log in from the terminal using the command

```
# pmadm -d -p ttymon0 -s a
```

This will display "`TERMINAL DISABLED.`" on the terminal.

A shell script that implements this sequence, add_ttyport, is available from the OPCOM FTP Server.

Adding Hardware After The Fact

Richard.Marejka@Canada.Sun.Com

With the introduction of Solaris 2.0, adding hardware after the installation procedure is not as easily accomplished as in SunOS 4.x. The kernel does not necessarily contain the driver and the filesystem may not contain the entry. For example, a system that has never had a tape drive attached it will have neither kernel nor filesystem support for the device.

There are two ways to add support for these missing devices. One way is to halt your system and reboot with the “-r” option, making certain to have the device powered on. For some users this is not an acceptable option.

An alternative is to execute `drvconfig` and one of `disks`, `ports` or `tapes` depending on the device being added. These programs can be found in the `/usr/sbin` directory. If the SunOS 4.x name space is required as well then `/usr/ucb/ucblinks` must be executed.

For example adding a tape would involve the following command sequence:

```
# /usr/sbin/drvconfig
# /usr/sbin/tapes
# /usr/ucb/ucblinks
```

As is the case with the “boot -r”, sequence the device must be attached to your system and powered on.

To confirm that the device is now present use the prtconf command. The driver should be present and a minor device number assigned (not -1).

Feature Articles for Next Month

User Level Threads
A Batch Scheduling Class
Porting NIT applications to DLPI
Loadable System Calls
Solaris 2.1 External Alpha

SunOpsis is published monthly by OPCOM.

To subscribe or contribute articles please contact:

Solaris 2.0 Support Centre
100 Renfrew Drive,
Markham, Ontario,Canada
L3R 9R6

1-800-363-6200 or email us at
solaris2@Sun.Com

Publisher: Dean Kemp
Editors: Brian A. Onn / Gil Hauer
Design & Layout: Lisa Thompson / Deborah Hummel
Contributing Writers:
Enrico Giuditti
David Guerra
Gil Hauer
Deborah Hummel
Dean Kemp
Richard Marejka
Georg Nikodym
Brian A. Onn
Gene Saunders
Gerry Singleton
Lisa Thompson
Larry Tsui
Srinath Venkat
Ron Winacott