Startup and Shutdown [4]

This section includes procedures to do the following:

- Start the CRAY J90se mainframe and bring up UNICOS to a multiuser run state mode (startup; also called *booting*).
- Bring UNICOS back to single-user mode (shutdown).

This section also briefly describes various aspects of single user and multiuser mode.

If you have access to a windowing environment, UNICOS provides a point-and-click, X Window System based interface to the UNICOS Installation / Configuration Menu System.

4.1 Related startup and shutdown documentation

The following documentation contains more detailed information about the material presented in this section:

- *UNICOS Administrator Commands Reference Manual*, Cray Research publication SR–2022
- *SWS-ION Administration and Operations Guide*, Cray Research publication SG–2204
- *SWS-ION Reference Manual*, Cray Research publication SR–2205
- *SWS-ION Release Overview*, Cray Research publication RO–5292
- *SWS Operating System and Devices Installation Guide*, Cray Research publication SG–5293

4.2 Booting the default configuration

It is recommended that you first boot the system with the default configuration. Later, you can make and test basic configuration changes, and finally configure file systems and make other site-specific changes. Using this approach can help isolate any errors that might be introduced during the configuration process.

Note: The CRAY J90se mainframe, SWS, and MPN must be correctly cabled in order to boot the CRAY J90se system. In addition, the SWS and MPN must be booted and running. For more information, see the *SWS-ION Administration and Operations Guide*, Cray Research publication SG–2204.

To boot your CRAY J90se mainframe over a GigaRing channel, execute the `bootsys`(8) command. The `bootsys` command boots one or more system components. A system component can be a ring, I/O node, or mainframe. If you do not specify any options, `bootsys` boots the I/O nodes, initializes the rings, and boots the mainframes (all the system components that are specified in the `topology`(5) file). The `topology` file identifies rings, I/O nodes, and mainframes and the manner in which they are connected. See the `topology`(5) man page for details. If you specify the `-c` option, a console is started for each mainframe that is booted.

Caution: Do not execute `bootsys`(8) on a running system. The system will crash, possibly causing a loss of data or corrupted file systems. Before executing this command, shut down UNICOS according to the methods described in UNICOS administrator documentation.

`bootsys` uses the defaults provided by the low-level commands, such as `bootj90`, unless they are overridden by specifications in the `options` file. For more information about the `options` file, see the `options`(5) man page.

If you specify one or more system components on the `bootsys` command line, only those components are booted. The order in which components are specified on the command line is insignificant.

For more information about the `bootsys`(8) command and about the SWS, see the `bootsys`(8) man page and the *SWS-ION Administration and Operations Guide*, Cray Research publication SG–2204.

Note: By default, the CRAY J90se system is booted in single-user mode. (The default mode, or *run-level*, is controlled by the `run_level` set on the UNICOS `inittab` `init` default line (normally S). At this point in the initialization process, the system should remain in single-user mode.

At this point, your system is running the default configuration.

Procedure 1: Booting your system

1. To boot your mainframe over a GigaRing channel, execute the `bootsys`(8) command at the prompt (if you specify the `-c` option, a console is started for each mainframe that is booted):

```
# bootsys -c
```

After executing the bootsys command, UNICOS is in single-user mode.

Only a few processes are running: init, swapper, idle, and sh. The root (/) file system is the only file system available.

When you are in single-user mode with only the root (/) file system available, you must do all editing by using the ed editor, because the vi editor is located in the /usr file system. If you want to use the vi editor before going to multiuser mode, you must mount the /usr file system. Before going to multiuser mode, or if you intend to work in single-user mode, you should check the root (/) file system by using the fsck command.

The first time you use the vi editor, you may see the following error message:

```
I don’t know what kind of terminal you are on - all I have is
’unknown’.[Using open mode]
```

Type the following command lines:

```
# TERM=xterm

(or sun-cmd, if you are using the command tool)

# export TERM
# resize
# echo $TERM
# echo $SHELL
```

2. Bring the system to multiuser mode by signaling the /etc/init process to change to a new run level by entering the following command:

```
# /etc/init 2
```

Multiuser mode is usually run-level 2. Although you can configure a system to run in multiuser mode at any level between 0 and 6, you may want to reserve some states for the future.

As the system boots into multiuser mode, output is produced on your terminal. You will be asked whether you want to run mkfs /tmp (y/n), which you must respond to for the process to proceed. At approximately midpoint in the process, the Administrative cleanup message appears. This message indicates that the system is moving into multiuser mode properly. You will be prompted for the system date, which is an optional entry. When the system boot is complete, you will see the following prompt:

```
Console Login:
```

4.2.1 Multiuser mode

Traditionally, run level 2 is the system's primary run level for multiuser mode. Among the initializations generally performed for multiuser mode are the following:

- Recording system start-up time in `/etc/wtmp`.
- Mounting all file systems required for normal system operation. This includes the regular system file systems (`/usr` and `/tmp`), the file system or systems that contain the home directories' `/tmp` file system of the system's users, and other file systems that contain files to which the users must have access.
- Removing any lock files that may interfere with normal system operation (for example, a lock file for a system daemon).
- Running daemons that provide various system services. The list may include, but is not restricted to, the following:
  - `errdemon`
  - `slogdemon` (for the UNICOS multilevel security (MLS) feature)
  - `cron`
  - `tapestart` (for online tapes)
  - `syslogd`
  - `nqsdaemon` (for NQS)
- Running the `netstart` script to initialize the system's TCP/IP network connections.
- Starting system accounting.
- Moving or truncating log files (for example, `/usr/lib/cron/log` or `/usr/spool/nqs/log`) to prevent them from growing without limits.
- Allowing users to log in.

4.2.2 Typical tasks you can perform while in multiuser mode

The following are some typical system administration tasks that you can perform while UNICOS is running in multiuser mode. The most important areas to monitor include how efficiently the system is performing and the rate at which system resources are being consumed.

- Checking which file systems are mounted by using the `/etc/mount` command.
- Checking all mounted file systems to ensure that no mounted file system consumes all available free disk blocks by using the `/bin/df` command or the `/etc/fsmon` file system monitor.
- Checking the number of system users by using the `who` command. To identify idle users, enter `who -u`. To determine the number of users, enter `who | wc -l`. To generate the number of users and a list of their names, enter `who -q`.
- Informing users of system changes by using `/etc/wall`.
- Monitoring how your UNICOS system is running by using the `/usr/bin/sar` utility. The `/usr/bin/sar`(1) utility has many options used to gain information about disk performance, character list buffers, CPU performance, and IOS throughput. The most useful options for a system administrator include `-d` (disk), `-x` (IOS), and `-v` (critical internal system table sizes). For more information, see the `/usr/bin/sar`(1) man page.
- Checking all running processes by using the `ps`(1) command to determine whether a process is using an abnormally large amount of CPU time. The `-eaf` options generate a full listing for all running processes.
- Checking the contents and size of your UNICOS error logs. Usually, error logs are found in the `/usr/adm` directory. Also, ensure that the error logging daemon is executing and that IOS disk errors are being logged into the `/usr/adm/errfile` file. For log information, see Chapter 9, page 179. For details on disk error reporting, see the `/etc/errpt`(8) man page.
- Checking `mail` by using the `/bin/mail` command while logged on to `root`, or the login that receives requests to restore files. If a problem occurs, the system itself sometimes sends mail to `root`.

4.2.3 Dedicated system

It is sometimes necessary to provide dedicated system time so that a particularly large or time-critical job can run unencumbered by other user processes. There also will be times at which system development work requires that the system be brought up as though it were running in multiuser mode, when access to the machine is actually restricted to the system staff. To lock out all users except yourself, use `/etc/udbrstrict -r -L` *your_userid*. Do not use just `/etc/udbrstrict -r`, because this limits logins to only root, which can then be done only on the console device. For more information about the UDB `ue_permbits` field, see *General UNICOS System Administration*, Cray Research publication SG–2301.

4.2.4 Run-level configuration

A *run level* is a software configuration of the system. Each run level allows only a selected group of processes to exist. Although run levels are most commonly used to configure the system in single-user or multiuser operation modes, thoughtful management of the run-level configuration on the system is a convenient method of tailoring the system’s resources to accommodate users’ needs.

Two main modes of operation exist for UNICOS: single-user and multiuser. Single-user mode is always indicated by run level s or S. Multiuser mode is typically run level 2, although it may be level 0 through 6.

One common use of the `/etc/inittab` file is to set up a run level so that certain procedures are followed automatically only the first time a run level is entered. For example, usually you are asked to verify the date and to check the file systems the first time you change your system to multiuser mode. These actions are caused by an entry in the `inittab` file. Subsequent changes in run level do not result in this procedure automatically unless you specifically change the `inittab` file.

4.2.4.1 Changing run level

As system administrator, you can change the run level by issuing the following command; *level* is the run level you want to initiate:

`/etc/init` *level*

The `/etc/inittab` file controls the specific actions that occur when a run level is initiated. The following subsections discuss the strategies for using run levels for various purposes.

4.2.4.2 Strategies for using run levels

Successful use of run levels requires that you think through the requirements for the system and tailor the initializations of the various run levels to provide for convenient transitions from one run level to another.

All systems have a single-user mode (for system work that must be performed unencumbered by the presence of other users on the system) and at least one multiuser mode. If the system is restricted at various times to dedicated use by one or more users, you should devote one or more run levels to initializing the system for this dedicated use. In all cases except for single-user mode (which requires little or no initialization), the `rc` (see the `brc`(8) man page) script performs initialization.

Procedure 2: Shutting down UNICOS

To shut down UNICOS, execute the following steps:

1. Make sure that you are logged in as `root` and that you are in the `root` (`/`), `/etc`, or `/ce` directory; to change to the `root` (`/`) directory, enter the following command:

   ```
   # cd /
   ```

2. You may want to send active users a special message about when the system will be shut down. The `/etc/shutdown` script is designed to return the UNICOS system to single-user run state in a clean, orderly manner. The `/etc/shutdown` script prompts you for a message that will be sent to all users; if you want to include a message, use the `wall`(8) command to provide the message (see section Chapter 8, page 173). Before executing `/etc/shutdown`, you can use the `ps -eaf` command to see processes that are running, and the `who -u` command to see whether people are actively using the system. The `shutdown`(8) command uses the following format:

   `/etc/shutdown` *grace-period-in-seconds*

   The following command instructs the system to wait 5 minutes (300 seconds) before terminating all processes and shutting down the system:

```
# /etc/shutdown 300
Do you want to send your own message?  (y or n):  y
Type your message followed by a <Return> and then ctrl d....
System shutting down in 5 minutes for test time-Please log out now.
CONTROL-d
```

The time it takes for the shutdown to complete depends on the number of processes that must terminate and file systems that must be unmounted; however, the shutdown process may take 3 to 5 minutes.

When the shutdown program is complete, the following message is displayed, and you should type the following highlighted commands:

```
Message: INIT: SINGLE-USER MODE.
# /bin/sync
# /bin/sync
# /bin/sync
# /etc/ldsync

(if you are using ldcache)

# df

(to verify that all file systems have been unmounted cleanly)

# /bin/sync
```

At this point, you are in single-user mode but UNICOS is still running. You can perform any system administration work as necessary.

4.2.4.2.1 Single-user mode

Many system maintenance, modification, testing, configuration, and repair procedures are performed while the system is in single-user mode to protect system users from potential instability and to ensure that user processes do not interfere with the system’s work while it is in progress. Therefore, the purpose of performing any initialization before the system is in single-user mode is to ensure that the system is known to be in an idle state.

When the UNICOS system is in single-user mode, all network connections and hard-wired terminals are disabled, and only the console terminal can interact with the system. This mode of operation lets you make necessary changes to the system without doing any other processing. When UNICOS is in single-user mode, the # symbol (or `sn`*xxx*#) is the system prompt.

Typically, the system is brought into single-user mode either following a system boot or by using the `shutdown`(8) command. In neither case should any user processes be running after the system is in single-user mode (no user processes will have started following a boot, and `shutdown` kills all user processes before entering single-user mode). Thus, there should be no need for initialization related to user processes when the system enters single-user mode.

As an extra measure of protection against inadvertent damage done to a mounted file system by single-user mode development work or testing, you should unmount all file systems except the current `root` file system. Traditionally, users doing the system work or testing while in single-user mode mount only the partitions they require. To help with this aspect of system work, you can provide a script in `/etc` that mounts the file systems that contain system commands not usually found on the root partition (the `/usr` file system) and the `home` user file system directories of the system staff.