2. Software Enhancements

This section describes the new features and enhancements included in the UNICOS 9.0 release. For additional information about upgrading to this release, see the description of compatibilities and differences in section 3.

Because this release overview documents all features introduced since the UNICOS 8.0 base release, each feature includes the UNICOS 8.0 release level in which the feature was introduced. This information is provided to help our customers focus on the features that are new specifically for their upgrade.

Each subsection in this section lists in the margin both the type of user and type of hardware affected. For definitions of the terms used, see subsection 1.6, page 1–6.

For information about UNICOS multilevel security (MLS) system enhancements and changes, see section 4.

Note: The documentation that supports the features listed in this section is available either in print or online. For a description of the documentation provided with this release, see section 5, page 5–1.

2.1 New CRAY T90 series supported

Initial release: UNICOS 8.3

The UNICOS 9.0 release includes support for the CRAY T90 series of mainframes. The CRAY T90 systems use new technology and system organization designed to deliver the benefits of system resiliency, flexible system configuration, and significant performance gains.

The CRAY T90 series is available in three models: the CRAY T94, the CRAY T916, and the CRAY T932 models. CRAY T94 systems have up to 4 CPUs and 128 Mwords of memory; CRAY T916 systems have up to 16 CPUs and 512 Mwords of memory; CRAY T932 systems have up to 32 CPUs and 1 Gword of memory.

CRAY T90 systems use an IOS model E (IOS-E), and the SSD solid-state storage device and peripherals supported on an IOS-E.

The following subsections provide detailed descriptions of features included in the UNICOS 9.0 release to support CRAY T90 systems.

2.1.1 *CRAY T90 basic support*

Users affected

All

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

Changes were made to the UNICOS operating system to provide basic control for the following hardware features of the CRAY T90 series:

- Exchange package format changes
- Size of A register increased to 64 bits
- Number of SB and ST registers increased to 16
- Number of semaphore registers increased to 64
- New logical operations on A registers
- Sparse channel numbering
- Number of channels increased
- Maximum number of CPUs increased to 32
- New Address Multiply Interrupt (AMI) interrupt type
- New I/O instruction for polling channel interrupt status
- New segmented address space hardware (LATs)
- New exchange address management
- New addressing and memory allocation granularity
- CRAY C90 compatibility mode
- Hardware performance monitor changes

Changes were also made in the UNICOS operating system to recognize the CRAY T90 system type, as follows:

- Serial number recognition:
  - 70*xx* are serial numbers for CRAY T94 systems.
  - 71*xx* are serial numbers for CRAY T916 systems.
  - 72*xx* are serial numbers for CRAY T932 systems.

- System target `cray-ts` was added, along with associated machine characteristics.

For more information, see the `target`(1) and `target`(2) man pages.

Fault recovery for CPU and memory-related faults is provided. New status register formats, memory discovery and probing, and new memory organization are all supported. Appropriate error collection, logging, and processing through `errdemon`(8) and `errpt`(8) are supported.

In CRAY T90 mode, the address multiply operands and result are 48 bits wide. If an overflow is detected on an address multiply operation, an Address Multiply Interrupt is generated and the process is sent the new `SIGAMI` signal. For more information on signals, see the `signal`(2) man page.

2.1.2 *CRAY T90 scalar cache support*

Users affected

All

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

CRAY T90 systems contain a 1024-word scalar cache in each CPU. Both native CRAY T90 mode and CRAY C90 compatibility mode can take advantage of the scalar cache.

UNICOS scalar cache support in the kernel provides control for enabling and disabling caching, and maintains cache coherency when asynchronous kernel services are provided to user processes.

Multitasked processes in CRAY C90 mode have scalar caching disabled by the kernel. The scalar cache, as a resource, can be disabled by using the `cpu`(8) command on a per-CPU basis. The `cpu`(8) command can also be used to force a command to execute with cache enabled or disabled.

For more information, see the `cpu`(4) and `cpu`(8) man pages.

2.1.3 *CRAY T90 shared memory*

Users affected

All

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

Support was added for the UNIX System V interprocessor communication (IPC) mechanism. This mechanism introduces three named object types to the UNICOS system: shared memory segments (CRAY T90 series only); semaphores; and message queues. For information about semaphores and message queues, see subsection 2.9.25, page 2–78.

The shared memory feature on CRAY T90 systems allows memory segments to be shared among cooperating processes. The following new system calls support use of shared memory segments:

System call	Description
`shmat`(2)	Attaches shared memory segment
`shmctl`(2)	Provides shared memory control operations
`shmdt`(2)	Detaches shared memory segment
`shmget`(2)	Accesses shared memory identifier

The following configuration parameters have been added to the UNICOS installation and configuration menu system in support of shared memory configuration: `SHMMAX`, `SHMMIN`, `SHMMNI`, and `SHMSEG`.

The new `crash`(8) subcommand `shm` displays information on shared memory segments.

The `shmctl`(2) call provides scalar cache support for shared memory processes. If the kernel list of processes caching each segment becomes corrupted, all processes with that segment attached are sent the new `SIGSMCE` signal. The default action is termination.

> **Note:** Jobs using shared memory segments cannot be checkpointed.

Changes to the following UNICOS subsystems support shared memory:

- Network Queuing System (NQS); see subsection 2.1.4, page 2–5, for information.
- User database (UDB); see subsection 2.1.5, page 2–6, for information.

- Multilevel security (MLS) feature; see subsection 4.2.8, page 4–5, for information.

For more information on this feature, see the following man pages: `ipcrm`(1), `ipcs`(1), `shmat`(2), `shmctl`(2), `shmdt`(2), `shmget`(2), `signal`(2), `stdipc`(3), `ipc`(5), `shm`(5), `ipc`(7), and `crash`(8).

Related publications

- *UNICOS Configuration Administrator's Guide*, publication SG–2303

2.1.4 *NQS shared memory supported*

Users affected

All

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

Users can specify the shared memory requirements of a Network Queuing System (NQS) batch request that will execute on a CRAY T90 series host. Specifying these resources lets administrators and operators manage the execution of these requests.

Users can now specify the following resources by using the `qsub -l` command option:

- `shm_limit` specifies the maximum size of shared memory created by the request.
- `shm_segments` specifies the maximum number of shared memory segments created by the request.

The `qlimit`(1) and `qstat`(1) displays have been enhanced to show the appropriate shared memory information.

The following subcommands were added to the `qmgr`(8) command to support shared memory resources:

- `set per_request shm_limit` queue sets the maximum per-request shared memory size limit for a batch queue.
- `set per_request shm_segments queue` sets the maximum per-request shared memory segment limit for a batch queue.

The `qmgr modify request` command has been enhanced to support modification of the shared memory resources specified by a request.

A request can specify shared memory resources when it is submitted at any UNICOS 9.0 host. The resource limits are valid only when the job executes on a CRAY T90 series host.

For more information, see the `qsub`(1), `qlimit`(1), `qstat`(1), and `qmgr`(8) man pages.

Related publications

- *UNICOS NQS and NQE Administrator's Guide*, SG–2305

2.1.5 *UDB supports CRAY T90 shared memory*

Users affected

Administrator

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

The user database (UDB) has two new fields to support shared memory on CRAY T90 systems: `jshmsegs` and `jshmsize`. For more information, see the `udbgen`(8) and `libudb`(3) man pages.

2.1.6 *CRAY T90 programming support*

Users affected

Administrator, programmer

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

The following operating system support has been provided for programming on CRAY T90 systems:

- The `sim`(1) command supports CRAY T90 system simulation. For more information, see the `sim`(1) man page.
- The system library (`libc`) supports CRAY T90 systems.
- The multiprocessing library and assembler definitions file (`asdef`) support CRAY T90 systems. These include changes to the context exchange package in `asdef` and the corresponding support changes in the multiprocessing library. This feature also includes macrotasking support for the CRAY T90 scalar cache.

- The Cray Assembly Language (CAL) assembler is supplied as an executable on CRAY T90 systems. For customers with sources licenses, the CAL source is not buildable. For nonsource customers, the CAL source is not viewable. The instruction set has been modified to run on CRAY T90 systems, and the new instructions that are necessary are provided. See the as(1) man page for more information.

Related publications

- *Cray Assembly Language (CAL) for Cray PVP Systems Reference Manual*, publication SR–3108
- *UNICOS Macros and Opdefs Reference Manual*, publication SR–2403

2.1.7 *Maintenance interfaces supported*

Users affected

Administrator

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

To support running the UNICOS kernel in CRAY T90 mode, the UNICOS system now provides the ability to evaluate and test a CPU or I/O port by locking the CPU or I/O port into maintenance memory, an area of memory protected from the remainder of memory and used for diagnostics.

The chmem(8) command now supports options that allow an administrator to display and configure the maintenance memory area. The mem(4) file can now be used to change the configuration of physical memory.

For more information, see the chmem(8) and mem(4) man pages.

2.1.8 *CRAY T90 support channel*

Users affected

Administrator

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

The CRAY T90 series has a direct low-speed channel connection between the mainframe and the support system (OWS and MWS). The support channel feature includes an E-packet simple protocol driver that manages packet traffic between the mainframe and the support system.

The support channel is used to permit TCP/IP network access to the OWS. The network connection provides session and file transfer support between the mainframe and the support system. It is also used during the installation process. The support channel is defined in the parameter file.

2.1.9 *CRAY T90 configuration tools on the support system (OWS)*

Users affected

Administrator

Supporting hardware

CRAY T90 series

Initial release: UNICOS 8.3

Configuration of CRAY T90 systems is performed with the aid of the following new software tools:

- `pact`(8) parameter configuration tool
- `fsct`(8) file system configuration tool
- `tconfig`(8) machine environment command

For more information, see the `fsct`(8), `pact`(8), and `tconfig`(8) man pages on the OWS.

Related publications

- *Support System Administrator's Guide*, publication SG–3079
- *Support System Reference Manual*, publication SR–3077

2.2 CRAY J90 series supported

Users affected

All

Supporting hardware

CRAY J90 series

Initial release: UNICOS 8.0.3.2J

Changes were made to the UNICOS operating system to add support for the new CRAY J90 air-cooled supercomputer, which includes an IOS-V. The CRAY J90 series uses standard 50/60 Hz, 200–240 V power, and UNICOS is bundled with it.

UNICOS software is preinstalled on a CRAY J90 system before the system is shipped to a customer. A window-environment tool provides a user-friendly interface for the installation of UNICOS upgrade releases and other Cray Research products.

Related publications

- *UNICOS Installation Guide for the CRAY J90 Series*, publication SG–5271
- *UNICOS Basic Administration Guide for CRAY J90 Series and CRAY EL Series*, publication SG–2416
- *CRAY IOS-V Commands Reference Manual*, publication SR–2170
- *CRAY IOS-V Messages*, publication SQ–2172

2.3 Support for new peripheral devices

The following subsections describe the new hardware, in addition to CRAY T90 and CRAY J90 systems, that is supported by the UNICOS 9.0 release.

2.3.1 *ESCON channel supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems with an IOS-E

Initial release: UNICOS 8.0.4/8.3

The tape subsystem in the UNICOS 9.0 release now supports the ESCON channel. When this channel is added to a tape configuration, it allows a greater number of tape devices to be attached to a Cray Research system and provides a higher bandwidth than is currently possible with a block multiplexer (BMX) channel.

The updated `tapeconfig`(5) file format, `tpbmx`(8) command, and `tpconfig`(8) command allow administrators to define ESCON interfaces. New error messages also have been added.

For more information, see the `tapeconfig`(5), `tpbmx`(8), and `tpconfig`(8) man pages.

The `esconmon`(8), `xesconmon`(8), `etf_escon`(8), and `elf`(8) IOS-E commands have also been updated to reflect ESCON channel support. For more information, refer to the man pages for these commands on the support system. The support system consists of the OWS, the MWS, and (if you have a CRAY T90 system) the CRAY T90 support multiplexer (TSM).

Related publications

- *UNICOS Tape Subsystem Administrator's Guide*, publication SG–2307
- *UNICOS Tape Subsystem User's Guide*, publication SG–2051
- *Support System Reference Manual*, publication SR–3077

2.3.2 *IBM 3495† and IBM 3494 autoloaders supported*

Users affected

End user, administrator, operator

Supporting hardware

All Cray Research systems except CRAY J90 series and CRAY EL series

Initial release: UNICOS 8.0.3/8.3

The UNICOS 9.0 software recognizes the IBM 3495† and 3494 Tape Library Dataservers as valid loader types. Each of these autoloaders is a linear, self-contained product that manages and automates the retrieval, mount, dismount, and storage of tape cartridges without the intervention of an operator.

The IBM 3495† Tape Library Dataserver uses tape drives from the IBM 3490 Magnetic Tape Subsystem and is available in several models. For all models, the minimum number of tape drives is four; the maximum number of drives differs for each model. For the largest model, the maximum number of tape drives is 64. The tape drives connect to Cray Research systems by using the existing block multiplexer channel or the ESCON channel on CRAY Y-MP systems with an IOS-E.

The IBM 3494 Tape Library Dataserver uses 3490E tape drive models C1A and C2A. These models work only with the IBM 3494 Tape Library Dataserver. The IBM 3494 Tape Library Dataserver is expandable in various ways; you can add on to the basic control unit with a combination of drive and storage units. The minimum number of tape drives is one; the maximum is eight.

For information about configuring this device, see the `tapeconfig`(5) man page.

† Implementation of IBM 3495 Tape Library Dataserver support is deferred.

This software requires a separate license from UNICOS 9.0. For information about licensing this software, see subsection 7.2.3.5, page 7–9, or contact your Cray Research representative.

Related publications

- *UNICOS Tape Subsystem Administrator's Guide*, publication SG–2307
- *UNICOS Tape Subsystem User's Guide*, publication SG–2051

2.3.3 *SI-3 controller supported on CRAY J90 series and CRAY EL series*

Users affected

End user, administrator, operator

Supporting hardware

CRAY J90 series and CRAY EL series

Initial release: UNICOS 8.0.3

CRAY J90 series and CRAY EL series now support the Small Computer System Interface (SCSI) tape controller type (SI-3). The SI-3 supports the fast and wide SCSI standards. The SI-3 can transfer data up to 20 Mbyte/s.

Related publications

- *UNICOS Tape Subsystem Administrator's Guide*, publication SG–2307

2.3.4 *DD-6S (9 Gbyte) disk drive support added for CRAY J90 series*

Users affected

End user, administrator, operator

Supporting hardware

CRAY J90 series

Initial release: UNICOS 8.0.3

The DD-6S (9 Gbyte, 5.25 inch) disk drive is now available for the CRAY J90 series. This drive can sustain transfer rates of 7 Mbyte/s for reads and 4 Mbyte/s for writes.

Related publications

- *UNICOS Basic Administration Guide for CRAY J90 Series and CRAY EL Series*, publication SG–2416

2.3.5 *Network Disk Array supported on CRAY J90 series and CRAY EL series*

Users affected

Administrator

Supporting hardware

CRAY J90 series and CRAY EL series

Initial release: UNICOS 8.0.3/8.0.4J

Network Disk Array devices (high-performance, high-capacity, disk storage devices) are supported on the CRAY J90 series and CRAY EL series. Support is provided by using the IPI-3 protocol over the HIPPI physical layer. The UNICOS driver, hdd, was modified to interact with the CRAY J90 or CRAY EL memory HIPPI driver. HIPPI connectivity allows configurations through HIPPI switches and HIPPI fiber-optic channel extenders. For more information, see the hdd(4) man page.

2.3.6 *CREMS/ER90 device supported on CRAY EL series*

Users affected

End user, administrator, operator

Supporting hardware

CRAY EL series

Initial release: UNICOS 8.0.3

The CRAY EL series now supports the ER90 helical-scan tape drive and its associated autoloader. All currently supported tape operations are supported on ER90 devices, and additional features unique to ER90 devices are provided for the first time on CRAY EL series systems. (ER90 hardware was previously supported only on CRAY Y-MP systems that had an IOS model E.)

The ER90 is a high-speed, helical-scan tape device. It uses a tape format based on the D-2 standard for digital composite video recording. The D-2 format consists of helical-scan tracks and longitudinal tracks. Data is recorded on the helical-scan tracks, and positioning information is stored on the longitudinal tracks. You can group multiple ER90 tape devices in a DataTower.

Tape operations unique to ER90 devices include multiple partitions (logical volumes) per physical volume, volume verification using a format identifier, non-BOT (beginning-of-tape) loads and unloads, and a fast-positioning feature.

Prior to UNICOS 9.0, users of the ER90 device product and of the ER90 device product had to special order these products. With UNICOS 9.0, both products are included in the UNICOS release materials.

The functionality is accessible only to licensed customers who have a separate Cray FLEXlm license key. For more information about licensing this product, see subsection 7.2.3.6, page 7–10, or contact your Cray Research representative.

Related publications

- *UNICOS Tape Subsystem Administrator's Guide*, publication SG–2307
- *UNICOS Tape Subsystem User's Guide*, publication SG–2051

2.3.7 *DD-302 disk device and DA-302 disk array support*

Users affected

End user, administrator, operator

Supporting hardware

All Cray Research systems with an IOS-E

Initial release: UNICOS 8.0.4

DD-302 disk devices and DA-302 disk arrays are now supported. These devices are the final generation of 3.5-in., IPI-2 disk devices to be supported on systems with an IOS model E. Performance and capacity are improved over the previous generation product, which is the DD-301/DA-301.

for information on the physical characteristics of DD-302 and DA-302 disk devices, see the `diskspec`(7) man page.

Related Publications

- *General UNICOS System Administration*, publication SG–2301

2.3.8 *ND40 network disk support*

Users affected

End user, administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.4

The UNICOS operating system now supports the ND40 network disk device. The ND40 is the next-generation network disk product based on the Maximum Strategy GEN-5 product. This product offers increased capacity, increased performance, and improved resiliency features over the previous models, ND12 and ND14.

The hddmon(8) command provides a display that includes an `els` selection, that allows you to monitor and control disk devices on CRAY J90 and CRAY EL systems. For information about the hddmon(8) command, see subsection 2.9.14, page 2–69.

This software requires a separate license from UNICOS 9.0. For information about licensing this product, see subsection 7.1.5.2, page 7–4, or contact your Cray Research representative.

2.3.9 *Support for IPI-3-attached disk and tape storage devices*

Users affected

Administrator, system analyst

Supporting hardware

CRAY J90 series and CRAY EL series

Initial release: UNICOS 8.0.4

UNICOS features have been added to support the IPI–3/HIPPI tape K-packet driver on CRAY J90 and CRAY EL systems. The hpi3(4) man page describes the format of the configuration file for this packet driver when it is used with CRAY J90 and CRAY EL systems.

When you use the mknod(8) command to configure hdd device nodes, you can now use the *iopath* and the *flags* fields to allow for dynamic HIPPI channel selection. The new implementation of these fields is described in the hdd(4) man page.

The hddmon(8) command provides a display that includes an `els` selection, that allows you to monitor and control HIPPI disk devices on CRAY J90 and CRAY EL systems.

The hpi3(4), `hpi3_config`(8), `hpi3_stat`(8), `hpi3_start`(8), hdd(4), and hddmon(8) man pages have been updated to document these features.

2.4 CRAY T3D series support enhancements

The UNICOS 9.0 release supports the UNICOS MAX 1.2.0.5 release. The following subsections describe enhancements made to the UNICOS 9.0 release to support CRAY T3D massively parallel processing (MPP) systems.

2.4.1 *CRAY T3D Phase II I/O enhancements supported*

Users affected

Administrator, programmer

Supporting hardware

CRAY T3D systems

Initial release: UNICOS 8.0.3/8.3

The UNICOS 9.0 release includes changes to support Phase II I/O enhancements on CRAY T3D systems and Cray Research host systems.

Phase II I/O allows a direct high-speed (HISP) channel I/O connection between a Cray Research host system I/O subsystem model E (IOS-E) and a CRAY T3D system I/O gateway (IOG). This allows additional CRAY T3D bandwidth for Cray Research host systems that have no additional Phase I I/O connections due to a limited number of CPUs or shared modules in the host system.

The `open`(2) and `fcntl`(2) system calls support a new status flag. The new `O_T3D` status flag lets you control memory usage when the HISP channel connects the IOS and CRAY T3D system. Setting this flag means that subsequent I/O is performed from the CRAY T3D memory, rather than from secondary data segments (SDS) or from the main memory of the Cray Research host system. For more information, see the `fcntl`(2) and `open`(2) man pages.

The `t3d` target value has been added to the `hcon`(8) utility on the support system (OWS). For more information, see the `hcon`(8) man page on the support system.

Current HISP channel targets now include `mainframe`, `SSD`, and `T3D`. The UNICOS parameter file has an added flag, `T3D`, within the `ios_e` section. The UNICOS Installation/Configuration Menu System item `UNICOS parameters for MPP` has a new submenu called `I/O gateway (HISP)`, which allows the user to input parameters for the Phase II connection.

For more information about administering Phase II I/O on a CRAY T3D system, see the *CRAY T3D Administrator's Guide*, publication SG–2507.

Related publications

- *UNICOS Configuration Administrator's Guide*, publication SG–2303

2.4.2 *CRAY T3D Phase III I/O enhancements supported*

Users affected

Administrator

Supporting hardware

CRAY T3D systems

Initial release: UNICOS 8.0.4

UNICOS now includes support for Phase III I/O enhancements on CRAY T3D systems and Cray Research host systems. In support of Phase III I/O, the UNICOS parameter file now has an added entry in the `ios_e` section, as follows:

```
ios_e {
    cluster 0 {
       miop; eiop 0; eiop 1; eiop 2; eiop 3;       channel
010 is hisp 0 to mainframe, mode YMP;
       channel 014 is hisp 1 to T3D      , mode c200d200;
  boot miop with "/home/ice-ows/cri/os/ios/iopmux";
       boot eiop 0 with "/home/ice-ows/cri/os/ios/eiop.comm";
       boot eiop 1 with "/home/ice-ows/cri/os/ios/eiop.hpi";
       boot eiop 2 with "/home/ice-ows/cri/os/ios/eiop.dca2";
       boot eiop 3 with "/home/ice-ows/cri/os/ios/eiop.dca3";

phase_III;
     }
}
```

The UNICOS Installation/Configuration Menu System menu item `UNICOS parameters for MPP` has a new submenu called `Phase III I/O gateway` that allows a user to input parameters for the Phase III connection. The following two entries are associated with this menu item:

`CLUSNUM` The number of the IOS cluster that has the Phase III connection (for example, 0).

`PHASE_III` Indicates if there is a connection [YES or NO].

UNICOS MAX support for Phase III I/O is planned for the UNICOS MAX 1.3 release.

Related publications

- *UNICOS Configuration Administrator's Guide*, publication SG–2303

2.4.3 *MPP system activity monitoring added*

Users affected

Administrator

Supporting hardware

CRAY T3D systems

Initial release: UNICOS 8.0.4/8.3

The new `mppview`(8) command displays the activity occurring on a CRAY T3D system attached to a Cray Research host; `mppview`(8) receives data from the `sam`(8) daemon on the local Cray Research host or on a remote host if specified. The tool uses display routines based on `curses`(3) for line mode terminals. For more information about `curses`-based displays, see the `csam`(8) man page.

A graphical user interface is available through the `xmppview`(8) command. The display graphically represents usage of processing elements according to criteria you select and gives system performance statistics in tables accessible through pull-down menus. `xmppview`(8) uses the X Window System and contains a tutorial. For more information about `xmppview`(8), see the `xmppview`(8) man page. For availability of `xmppview`, contact your Cray Research service representative or Software Product Support.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.5 Network connectivity and communication enhancements

The following subsections describe enhancements made to the UNICOS 9.0 release to improve Cray Research systems network connectivity and communications.

2.5.1 *NQX 2.0 (NQE on UNICOS) supported*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

> **Note:** The initial NQX support was provided with the UNICOS 8.0.3 and 8.3 releases. Until the UNICOS 9.0 release, NQX 1.1 was supported and available with UNICOS.

The Network Queuing Environment (NQE) is a batch queuing system that automatically balances job loads across heterogeneous systems on a network. NQE contains the Network Queuing System (NQS), the File Transfer Agent (FTA), and the Network Queuing EXtensions (NQX) components. NQS and FTA are bundled with UNICOS and provide basic NQS and FTA services for UNICOS systems.

NQX adds components and functionality beyond NQS and FTA, such as a collector that provides system load and job status information to a Network Load Balancer (NLB) database. NQX also supports NQE clients on the server side and lets you take advantage of other NQE features, such as load balancing, application programming interfaces (APIs), and job dependency.

NQE on UNICOS systems consists of the following components:

- NLB (part of NQX)
- NQE client (part of NQX)
- NQS (bundled with UNICOS)
- FTA (bundled with UNICOS)

The following enhancements are included in NQX 2.0:

- The job dependency feature lets users or operations staff specify which job events should be posted and which job events should wait. The mechanism is implemented separately from NQS, which lets both NQS jobs and other applications use this batch complex event-posting mechanism.

- Support for submitting jobs as an alternate user has been extended. If you know the username/password pair, you can now run jobs as different users on any NQX server on which you have authorized access.
- Additional data objects in the NLB provide more information about NQE servers and jobs.

Cray NQX 2.0 is not supported on systems that have enabled UNICOS multilevel security (MLS). For more information, see subsection 4.3.2, page 4–10.

This software requires a separate license from UNICOS 9.0. For information about licensing this product, see subsection 7.2.3.1, page 7–8, or contact your Cray Research representative.

Related publications

- *UNICOS NQS and NQE Administrator's Guide*, publication SG–2305
- *NQE User's Guide*, publication SG–2148
- *NQE Ready Reference*, publication SQ–2149
- *NQE Administration*, publication SG–2150
- *Introducing NQE*, publication IN–2153

2.5.2 *NQS/URM scheduling*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

In UNICOS 8.0, additional NQS scheduling modes were introduced. In UNICOS 9.0, the description of these commands has been enhanced to provide more guidance on how to use the various options.

With the Unified Resource Manager (URM) scheduling modes, NQS works with URM before initiating jobs. URM scheduling is available only on UNICOS systems. To use URM for job scheduling, you must be running URM. By default, URM listens on port 606.

The `qmgr set job scheduling` options are as follows:

Option	Description
`nqs normal`	Traditional NQS scheduling; the default scheduling type. NQS does not use URM. It is the only mode that uses the NQS scheduling weighting factors.
`nqs resource_management`	Combines traditional NQS scheduling (`nqs normal`) with SDS management. It uses URM instead of the `qfdaemon` process, which is no longer available, and URM manages SDS preemption for NQS jobs.
`nqs superqueues`	This is `nqs normal` scheduling, with the following addition: jobs in batch queues of equal priority are scheduled as though they are members of a common queue. Individual queue limits continue to be enforced.
`urm limited`	Provides NQS limits coupled with URM scheduling. In this mode, URM determines which jobs will run, but it stays within NQS limits. This option is useful for sites that need to use their current NQS-style queue and scheduling structure, but want URM to take a more active part in job initiation.
`urm unlimited`	NQS registers all incoming jobs with URM and allows URM to make job initiation decisions. NQS starts all jobs as recommended by URM. URM evaluates its target load by examining the system load, including all work on the system. This includes non-NQS work, such as interactive sessions. As a result, NQS may start jobs in excess of NQS limits if it is recommended by URM.

For additional information about configuring and running the UNICOS URM, about URM targets and NQS limits, and for an explanation of the additional parameters URM uses in its job initiation logic, see *UNICOS Resource Administration*, publication SG–2302, or the NQS `qmgr` online help information.

For compatibility information, see subsection 3.4.1.4, page 3–10, and subsection 3.4.1.5, page 3–10.

2.5.3 *NQS superqueues*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

This release adds the new `Nqs Superqueues` job scheduling type to the `qmgr set job scheduling` command. This option lets you schedule jobs that are in queues of equal priority as though they are members of a common superqueue. For more information, see the `qmgr`(8) man page or the `qmgr` online help information.

Related publications

- *UNICOS NQS and NQE Administrator's Guide*, publication SG–2305

2.5.4 *Kerberos RPC server for multiple IP addressed hosts now supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.3/8.3

This UNICOS release supports an optional Remote Procedure Call (RPC)-based server, `krbipd`(8). For Cray Research systems that have more than one Internet Protocol (IP) address configured, `krbipd`(8) checks possible mismatched IP interface addresses against the list of IP addresses configured for a specific machine. For more information, see the `krbipd`(8) man page.

Related publications

- *Kerberos Administrator's Guide*, publication SG–2306

2.5.5 *Kerberized* `ftp/ftpd` *security extensions*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

Kerberized `ftp/ftpd` implements the `ftp`(1B) security extensions documented in the Internet draft document *FTP Security Extension*, April 1994. This feature extends `ftp`(1B) and `ftpd`(8) to provide strong authentication, integrity, and confidentiality on the control and data channels by introducing new, optional commands, replies, and file transfer encodings.

For more information, see the `ftp`(1) and `ftpd`(8) man pages.

Related publications

- *Kerberos Administrator's Guide*, publication SG–2306
- *Kerberos User's Guide*, publication SG–2409

2.5.6 *Cray DCE Client Services and Cray DCE DFS Server support*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.3

This UNICOS release supports the Cray Distributed Computing Environment (DCE) Client Services and the Cray DCE Distributed File System (DFS) Server. DCE is a set of services and tools that provide for distributed computing in a heterogeneous computing environment. The Cray DCE Client Services product provides client services such as Call Directory Services (CDS), Remote Procedure Call (RPC), Security Services, and Threads. The Cray DCE Distributed File System (DFS) Server product allows the UNICOS system to act as a DFS file server.

For information about checkpointing and restarting DFS files, see subsection 2.7.6, page 2–42. For information about DFS/SFS enhancements, see subsection 2.9.21, page 2–75. For information about `crash`(8) enhancements for DFS, see subsection 2.9.11, page 2–66. For DCE login differences with this UNICOS release, see subsection 3.4.5, page 3–12.

This software requires a separate license from UNICOS 9.0. For information about licensing these products, see subsection 7.1.5.2, page 7–4, or contact your Cray Research representative.

Related publications

- *Cray DCE Client Services 1.0.2.1 / Cray DCE DFS Server 1.0.3.1 Release Overview*, publication RO–5225

2.5.7 *File Transfer Agent (FTA) IBM block mode support enhanced*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.3/8.3

IBM MVS support was added to File Transfer Agent (FTA) functionality. Users can now transfer binary-blocked UNICOS files to and from record-oriented MVS files for use on IBM platforms. For more information, see the `fta`(8), `fta.conf`(5), and `ftua`(1) man pages.

Related publications

- *FTA User and Administrator Manual*, publication SG–2144

2.5.8 *File locking supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Remote file systems now support file locking, provided the new daemons `lockd`(8) and `statd`(8) are running. However, file systems can still be mounted without file locking.

The `lockd`(8) daemon processes file lock requests. The `statd`(8) daemon provides crash and recover functions for NFS locking services. For more information, see the `lockd`(8) and `statd`(8) man pages.

Related publications

- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.9 `gated` *OSPF supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

A new version of the `gated`(8) daemon supports the Open Shortest Path First (OSPF) routing protocol as well as a new version of the `ripquery`(8) command. The 9.2.1 software version of Cisco routers is interoperable with the OSPF protocol.

The following commands have been added:

- `gdc`(8) provides a user-oriented interface for the operation of `gated`(8).
- `ospf_monitor`(8) queries OSPF routers to provide detailed information on I/O statistics, error logs, link-state databases, autonomous system (AS) external databases, the OSPF interfaces, and OSPF neighbors.

For more information, see the `gated`(8), `gdc`(8), and `ospf_monitor`(8) man pages. For important compatibility information, see subsection 3.4.1.7, page 3–11, and subsection 3.4.4, page 3–12.

Related publications

- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.10 *New options for* `route` *command support* `gated` *routing daemon*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The following options and directives have been added to the `route`(8) command to support the `gated`(8) routing daemon:

Option	Definition
`-t`	Processes routing directives, but makes no changes to the kernel routing table
`-d`	Specifies debug mode, which prints the route message that would have been sent to the kernel when manipulating the route tables manually

Directive	Definition
`change`	Changes the gateway or metrics of a route
`get`	Looks up and displays the route for a destination

Also, the `-C` option has been removed from the `route`(8) command. For more information on this removal, see subsection 3.4.3, page 3–12.

For more information about this enhancement, see the `route`(8) man page.

Related publications

- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.11 `spnet` *command enhanced*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Enhancements were made to the `spnet`(8) command, including a new system call, `nsecctl`(2), to improve the performance, maintainability, extensibility, and user interface of `spnet`(8). Now, a single network access list (NAL) or workstation access list (WAL) definition can have several addresses associated with it in the kernel as well as in the `spnet.conf` file, resulting in potential memory savings in the kernel.

The `spnet.conf` file has changed. In the UNICOS 9.0 release, the `spnet`(8) command reads the `spnet.conf` file more efficiently, resulting in a reduction in system time equivalent to approximately 30% of total execution time. In addition, the `name` field declared in the `spnet.conf` file is loaded into the kernel for administrator convenience.

The following are new options in `spnet`(8):

- `-d[efault]` lists all the NAL field values, including defaults.
- `-n[umeric]` lists Internet Protocol (IP) addresses in dotted-decimal form.

For more information, see the `nsecctl`(2) and `spnet`(8) man pages.

Related publications

- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304
- *General UNICOS System Administration*, publication SG–2301

2.5.12 *IP multicasting support added*

Users affected

Administrator, programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release supports multicasting, which enables an application to send a single packet to a group of hosts. The new `igmp`(4P) protocol is used by hosts and routers to identify the multicast groups to which a host currently belongs.

Multicast routers use this information to forward multicast datagrams to the appropriate interfaces. For information about the new IP-level multicast options added to the protocol, see the `ip`(4P) man page.

A new IP multicast routing daemon, `mrouted`(8), forwards a multicast datagram along the shortest path tree that is rooted at the subnet on which a datagram originates. The following administrative commands were added:

- `map_mbone`(8) obtains routing information from multicast routers and builds a topological map from the information.
- `mrinfo`(8) obtains routing information from a multicast router and prints a description of the router and the virtual interface list.

For more information, see the `igmp`(4P), `mrouted`(8), `map_mbone`(8), and `mrinfo`(8) man pages.

Related publications

- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.13 `rcp` *command enhanced*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The `rcp`(1) command has been enhanced in functionality and performance.

The `-T` option lets users display transfer times and rates.

To provide better performance over HIPPI connections, the `rcp`(1) command now uses asynchronous double-buffered disk I/O and larger network buffer sizes.

For more information, see the `rcp`(1) man page.

Related publications

- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.14 *BIND 4.9.3.beta9 supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release supports Version 4.9.3.beta9 of the Berkeley Internet name domain (BIND). Updated commands include `dig`(8), `host`(1B), `named`(8), `named-xfer`(8), `nslookup`(1), and the resolver library. Additional debugging aids have been added to the resolver library. ANSI C compatibility and portability are enhanced.

The `named`(8) daemon provides negative caching of invalid names and addresses, and improved error checking. The new resolver library supports multiple default domains, address sorting, and improved error handling.

For more information, see the `dig`(8), `host`(1B), `named`(8), `named-xfer`(8), `nslookup`(1), `gethost`(3), `resolver`(3), `masterfile`(5), `named.boot`(5), and `resolv.conf`(5) man pages.

Related publications

- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.15 `crash` *enhancements for NFS*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3/8.0.3

The `crash`(8) administrator command has been enhanced to display information about the following networking features:

- Network file system (NFS) client handles
- NFS client handle tables
- NFS `svc_data` structures
- Remote Procedure Call (RPC) `SVCXPRT` structures

The following new subcommands support these features:

- `ch`
- `chtable`
- `svc_data`

- `svc_xprt`

For more information, see the `crash`(8) man page.

2.5.16 *Asynchronous Transfer Mode (ATM) technology supported*

Users affected

All

Supporting hardware

Cray Research systems with an IOS-E and HIPPI channel (BBG); CRAY EL and J90 series (Native ATM)

Initial release: UNICOS 8.0.4

Asynchronous Transfer Mode (ATM) is transmission technology that allows voice, data, and video transmission across a single connection. ATM technology offers a universal network connection for supercomputers, and also enables new applications for multimedia user interfaces, geographic independence of client-server resources, and ease of relocatable network modes. The Cray Research product name *ATM* is derived from and is used interchangeably with the transmission technology Asynchronous Transfer Mode.

Because off-the-shelf ATM hardware components do not fit into traditional Cray Research architectures, Cray Research has developed an external network adapter, a Bus Based Gateway (BBG), for these systems. CRAY EL series systems and CRAY J90 series systems use a traditional VME interface for ATM technology.

Man pages for the BBG interface are as follows: `bbg`(4), `bbgatmstat`(8), `bbgconfig`(8), `bbghwconfig`(8), `bbgoc3config`(8), and `bbgstart`(8).

Man pages for the VME interface are as follows: `atmarp`(8) and `atmadmin`(8).

ATM support for the CRAY EL series is available with the UNICOS 8.0.4 release and later.

For more information about this product, contact your Cray Research representative.

Related publications

- *Asynchronous Transfer Mode (ATM) Administrator's Guide*, publication SG–2193

2.5.17 *ONC+ technology supported*

Support for ONC+ technology has been added, as described in the following subsections.

For information about ONC+ licensing or to purchase an ONC+ license, see subsection 7.2.3.2, page 7–9, or contact your Cray Research representative.

2.5.17.1 *Network Information Service Plus (NIS+) supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

Network Information Service Plus (NIS+) is a new network information service that enables administrators to administer domains and network information access more efficiently. This service stores and manages administrative information required by programs, such as searching for a password, validating a user ID, or finding the address of a server host. It is one of SunSoft's Open Network Computing Plus (ONC+) technologies that is now available in the UNICOS system. The advantages of NIS+ include the following:

- Distributed and remote administration of network domains
- Support of multiple administrative domains
- Automatic updates of network information from master and slave servers
- Easier access to naming service information
- User authentication and permission control of NIS+ information

The `nis`(4) man page provides a brief overview of the product.

Cray Research continues to provide the Network Information Service (NIS) product with the UNICOS operating system under the UNICOS set license.

Related publications

- *ONC+ Technology for the UNICOS Operating System*, publication SG–2169
- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.17.2 AUTH_KERB *RPC authentication supported*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

AUTH_KERB is a new Remote Procedure Call (RPC) authentication that uses MIT's Kerberos Version 4 authentication mechanism. AUTH_KERB RPC technology enables programmers to write client-server applications that can securely request services on remote servers over untrusted networks.

AUTH_KERB RPC authentication is hardly noticeable by the end user. End users must only request a Kerberos ticket-granting ticket to access a Kerberos RPC server and Kerberized RPC applications.

For more information about remote RPC authentication, see the `kerberos_rpc`(3) man page.

Related publications

- *Kerberos User's Guide*, publication number SG–2409
- *ONC+ Technology for the UNICOS Operating System*, publication SG–2169
- *Remote Procedure Call (RPC) Reference Manual*, publication SR–2089

2.5.17.3 AUTH_KERB *Kerberized NFS supported*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

AUTH_KERB NFS is a new feature that uses MIT's Kerberos Version 4 authentication mechanism for network file system (NFS) protocol transactions. The UNICOS kernel calls a user level daemon, `kerbd`(8), to generate and validate Kerberos RPC requests.

The following example of a Kerberized NFS transaction between client and server using the change directory command, cd(1), illustrates the use of AUTH_KERB.

1. A client makes an NFS request to obtain access to a mounted file system with the cd(1) command. The UNICOS kernel on the client machine makes a request to kerbd to get Kerberos credentials on behalf of the user. Having obtained and encrypted those credentials, the UNICOS kernel makes a Kerberized (AUTH_KERB) RPC call to the server machine with a request for access.
2. The UNICOS kernel running on the server machine receives the request. It calls the kerbd daemon on the server to decrypt and verify the the AUTH_KERB RPC request. If the request is decrypted successfully, it is passed to the server's nfsd(8) daemon and serviced. nfsd is unaware that the message came in using AUTH_KERB RPC.

For more information about AUTH_KERB NFS, see the exportfs(8), kdestroy(1), kerbd(8), klist(1), mount(8), and nfsd(8) man pages.

Related publications

- *ONC+ Technology for the UNICOS Operating System*, publication SG–2169

2.5.17.4 *New version of NFS supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

NFS version 3 is a new version of the network file system (NFS), a protocol and distributed file system. It supports larger files, and allows improved permission checking and computing environment scalability. NFS code version 3 contains a revised collection of procedures, but it can run concurrently with NFS version 2. NFS version 3 contains a new file system type named NFS3.

NFS version 3 is largely transparent to UNICOS users and administrators. With this new protocol and file system type, users share enhanced file system access across large and complex networks.

Using the enhanced functionality of NFS version 3 and other components of Open Network Computing Plus (ONC+) technology, customers can link Cray Research machines with other machines in multiplatform enterprise networks.

The new NFS3 file system type supports file locking. If your system is licensed for ONC+, the `lockd`(8) daemon registers for version 4 of the `nlockmgr` protocol.

For additional information, see the `automount`(8), `lockd`(8), and `mount`(8) man pages.

Related publications

- *ONC+ Technology for the UNICOS Operating System*, publication SG–2169
- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304

2.5.17.5 *New RPC program number mapper* `rpcbind`(8) *supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

This UNICOS release includes the new `rpcbind`(8) daemon, which maps universal addresses to RPC program numbers and specific ports.

The `rpcbstart`(8) daemon starts the `rpcbind` daemon. If your site does not have an ONC+ license, the `rpcbstart`(8) daemon starts the `portmap`(8) daemon by default.

For more information, see the `rpcbind`(8) and `rpcbstart`(8) man pages.

Related publications

- *Remote Procedure Call (RPC) Reference Manual*, publication SR–2089

2.5.18 *NFS ID mapping conversion*

Users affected

Programmer, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The network file server (NFS) ID mapping domains have been converted to a radix tree structure. The benefits of this feature are as follows:

- The radix tree structure for NFS ID mapping domains permits faster worst-case lookups because the maximum lookup time is based on the depth of the radix tree, and there is a known maximum depth to the tree. The previous structure was a linked list, so the look-up operation was linear; this was potentially much slower (because the length of the list is limited only by kernel mbuf memory).
- Commands that are used to add ID mapping domains to the kernel are now order-independent.

2.5.19 *Increased network connectivity on CRAY J90 and CRAY EL systems*

Users affected

Administrator

Supporting hardware

CRAY J90 systems and CRAY EL systems

Initial release: UNICOS 8.0.4

To support greater connectivity, the maximum number of Ethernet and FDDI interfaces (controllers) on CRAY J90 systems and CRAY EL systems has been increased from 4 Ethernet and 2 FDDI interfaces to 8 Ethernet and 8 FDDI interfaces. The limit on the number of controllers on a CRAY EL IOS or on a CRAY J90 IOS-V remains at 4 Ethernet and 2 FDDI controllers per IOS.

2.6 User interface, tools, and utilities enhancements

The following subsections describe enhancements to the user interface, tools, and utilities in the UNICOS 9.0 release.

2.6.1 *New* `chkpnt_util` *utility added to aid in determining job restartability*

Users affected

End user, administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The `chkpnt_util`(1) utility has been added to allow users to examine a restart file. The utility performs most of the tests that are performed within the kernel and provides specific information about the cause of a restart failure. The utility also can copy a restart file and add names for data files referenced within the file.

For more information, see the `chkpnt_util`(1) man page.

2.6.2 *New search options added to* `man` *utility*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The `-s` and `-l` options were added to the `man`(1) utility. Descriptions of the new options follow; the new search options make the `man` *section title* option obsolete. The `man` *section title* option will be removed in a future release.

Option	Description
`-s` *section*	Specifies sections of the directory for `man` to search. The directories searched for title(s) are limited to those specified by *section*(s). *section* is an Arabic number or one of the words `new`, `local`, `old`, `public`, or `quick` (added for `-s` option, not available with obsolete usage `man` *section title* ...). A section may be followed by a one-letter classifier. To specify multiple sections, separate sections with a comma.
`-l` *title*	Lists all pages found matching the *title*(s) given in the search path. The search path is a colon-separated list of directories in which man page subdirectories may be found.

2.6.3 *New option to* `tar` *utility for file extraction with archived permissions*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

A new option has been added to the `tar`(1) utility. The `-p` option restores archived files with the original permissions preserved. When `-p` is specified, the current `umask` of the user is ignored, and the files are extracted with the permissions with which they were archived.

For more information, see the `tar`(1) man page.

2.6.4 `ticksum` *utility added*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The new `ticksum`(1) utility provides a time-independent checksum of a file.

If invoked with one argument, `ticksum`(1) calculates a checksum of the named file and writes this sum, in hexadecimal format, to standard output. For the specific kinds of binary files `ticksum`(1) recognizes, information such as the time of compilation and the version of the generating product is ignored in calculating the checksum.

If `ticksum`(1) is invoked with two arguments, `ticksum`(1) compares the two files by comparing their checksums. A zero exit status indicates that the checksums were the same; a nonzero exit status indicates that the checksums differed.

For more information, see the `ticksum`(1) man page.

2.6.5 *GNU Emacs upgraded to version 19*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Full X Window System support is now provided for the GNU Emacs editor (version 19.28). This version provides menu support for the X Window System and adds the `ispell`(1) spell checking function. For more information, see the `/usr/lib/emacs/README2` file. For information about important differences between the previous and the new versions of Emacs, see subsection 3.5.11, page 3–22.

2.6.6 `define` *utility added to Docview*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Support for the `define`(1) utility has been added to the X Window System version of Docview. This capability allows a user to obtain the definitions of Cray Research and site glossary terms from within Docview. This feature depends on the UNICOS `define`(1) utility being installed.

For more information, see the `docview`(1) and `define`(1) man pages.

2.6.7 `help` *command improved*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The `help`(1) command has been modified so that when it cannot find help for a specified topic, it executes the `explain`(1) or `man`(1) command to offer further assistance in defining keywords.

For more information, see the `help`(1) man page.

2.6.8 *PE time limit options added to* `limit` *command*

Users affected

End user

Supporting hardware

All Cray Research systems using MPP hardware

Initial release: UNICOS 9.0

The `limit`(1) command has been modified so that a user does not need to write a program to change processing element (PE) time limits. A user can now change PE time limits on the command line by using the `-t pe_time_limit` and `-e pe_limit` options.

For more information, see the `limit`(1) man page.

2.6.9 *UNICOS source manager (USM) enhanced*

Users affected

Programmer, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Several enhancements have been made to the UNICOS source manager (USM). A summary of the changes is as follows:

- Increased support for sideline branches allows for branches to be named.
- A new command, `unget`(1), was developed to replace the functionality of the `get -u` subcommand.
- A new program library (PL) type was added to allow additional sequence numbers per mod ID, as well as increasing line lengths to 2047 characters. This new PL type cannot be processed by earlier versions of USM, but specifying the `-o` option on the `create` subcommand line will create an *old type* PL. This new version of USM can process either type of PL.
- A more robust recovery mechanism was added to make recovery safer.
- Error messages were reorganized.
- The `bmerge` subcommand was added to the UNICOS source manager (USM) to merge sideline branches into the main line. The main purpose of this subcommand is to facilitate the merging of site-created local sideline branches back into the main line after the main line has been updated. The local sideline branch can then be built from the new main-line node.

  For more information, use the `sm`(1) `help` subcommand.
- The `-m` option was added to the `sm`(1) command line to allow suppression of informational, caution, or warning USM messages.
- The `-f` option has been added to the USM `get` subcommand to force the return of a file even if a writable copy exists in the user's directory.
- The addition of three options to the `sm`(1) command: `-a` lists all defined attributes; `-n` lists all named branches; and `-x` *ids* exits if all IDs are known.

- The ability to set time-based attributes and named branches; query whether attributes or named branches are known within a PL; and see all mod IDs that occurred between two attributes.

For more information, see the get(1) and sm(1) man pages.

2.6.10 bc *utility from the Free Software Foundation supported*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The bc(1) utility uses source code ported from the Free Software Foundation (FSF). The FSF bc(1) utility replaces the standard UNIX bc(1) utility issued with previous UNICOS releases because the FSF utility provides reliable calculations. Existence of problems in many implementations of the UNIX bc(1) utility is documented and relatively widely known.

The bc(1) and the dc(1) utilities included in previous UNICOS releases were renamed to be obc(1) and odc(1), respectively, in the UNICOS 9.0 release.

For information about the differences between the FSF and UNIX bc utilities, see subsection 3.5.6, page 3–19.

For more information on the FSF utility, see the bc(1) man page.

2.6.11 *New* whoami *utility supported*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release provides the new whoami(1B) utility. This utility displays the login name that corresponds to the current effective user ID. If you use su(1) to adopt another user temporarily, whoami(1B) reports the login name associated with that user ID.

For more information, see the whoami(1B) man page.

2.6.12 *Standard shell* `setuid/setgid` *script conformance supported*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The standard shell (a POSIX compliant version of the Korn shell) is now consistent with other UNICOS shells and does not honor the `setuid` and `setgid` modes of shell scripts. As a result of this change, the standard shell does not execute shell scripts that do not begin with a line reading `#!` and that are not readable.

For more information, see the `sh`(1) man page. For compatibility information, see subsection 3.5.2, page 3–17.

2.6.13 *New* `tpmnt`(1) *command* `-l` *option argument examines tape contents*

Users affected

End user, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The `tpmnt`(1) command provides a new argument, `ulp`, for the `-l` option. This argument is used primarily by the `tplist`(1) command to examine the contents of a tape, and it eliminates the need for `tplist`(1) to use `blp`.

For additional information, see the `tpmnt`(1) man page. For additional tape subsystem enhancements, see subsection 2.9.16, page 2–70.

2.7 System calls enhancements

The following subsections describe system call enhancements in the UNICOS 9.0 release.

2.7.1 *New* `unlink2` *system call*

Users affected

System analyst, programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.4

The new `unlink2`(2) system call functions like the `unlink`(2) system call with the exception of the return values. Both system calls remove directory entries. In addition, the `unlink2`(2) system call returns the number of bytes of space freed when all links are removed from a file.

For more information, see the `unlink`(2) man page.

2.7.2 *New inode-based system calls for locating files and directories*

Users affected

System analyst, programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.4/8.3

Two new system calls are available for locating files and directories. The `openi`(2) system call opens a file by using the inode number. The `chdiri`(2) system call changes to a directory by using the inode number.

For more information, see the `openi`(2) and `chdiri`(2) man pages.

2.7.3 *New* `lsetattr` *system call*

Users affected

System analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The new `lsetattr`(2) system call improves the speed and functionality of a system restore. The `lsetattr`(2) system call makes available the full generality of the `VOP_SETATTR()` macro.

For more information, see the `lsetattr`(2) man page.

2.7.4 `access` *system call using* `euid` *supported*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Programmers can access the effective user and group IDs by specifying the new `EUID_OK` bit in an `access`(2) system call. With the `EUID_OK` bit specified, the ID substitution (in which the real user and group IDs replace the effective user and group IDs) does not occur.

For more information, see the `access`(2) man page.

2.7.5 `signobdm` *signal masked*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release has been enhanced to allow a programmer to mask the `signobdm` signal in an application and thus block the signal from delivery. However, if an exception signal (such as `signobdm`, `sigfpe`, `sigore`, and so on) is masked, the application may continue to run even though it is failing, making fault isolation difficult. This change makes `signobdm` consistent with other signals.

For more information, see the `signal`(2) and `sigprocmask`(2) man pages.

2.7.6 `chkpnt` *and* `restart` *system calls support DFS files*

Users affected

End user, administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

Processes with open files that reside on Distributed File System (DFS) file systems can be checkpointed and restarted. Credentials would be lost across a system reboot, however.

For additional information, see the `chkpnt`(2) and `restart`(2) man pages.

2.8 UNICOS library enhancements

The following subsections describe library enhancements in the UNICOS 9.0 release.

2.8.1 *String-to-float conversion routine added*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

A new string-to-float conversion routine, `strtof`(3), converts strings to the `float` data type. This is most useful on Cray MPP systems where `float` is a 32-bit data type but `double` is 64 bits.

For more information, see the `strtod`(3) man page.

2.8.2 *User exit available at program start-up*

Users affected

Programmer, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

A user exit has been added to the program start-up routine `$START$`. This user exit gives anyone the ability to link a user-specified routine into a program.

As the last step before calling the main routine of the program, the `$START$` routine checks for the existence of a routine called `sitelocal_start`; if this routine is linked into the program, it is called. This allows site-specific user exit code to be run before the main routine is called. To use this feature, write a `sitelocal_start` function and link it to your program.

For more information and an example of a `sitelocal_start` function, see the `start`(3) man page.

2.8.3 *New routines for word-oriented memory operations*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

UNICOS 9.0 makes available three new routines for word-oriented memory operations: `memwcmp`(3), `memwchr`(3), and `memstride`(3).

For more information, see the `memword`(3) man page.

2.8.4 *Memory manager features*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Several requested memory manager features have been implemented in the UNICOS 9.0 release as follows:

- Initial release: UNICOS 8.3

  The following new routines have been added: `malloc_check` (added in UNICOS 7.0), `malloc_inplace`(3), `malloc_expand`(3), `malloc_extend`(3), `malloc_howbig`(3), `malloc_isvalid`(3), `malloc_space`(3), `malloc_limit`(3), `malloc_stats`(3), `malloc_tron`(3), `malloc_troff`(3), `malloc_etrace`(3), and `malloc_dtrace`(3).

  These routines are similar to the old routines formerly documented on the `memmgr`(3) man page. For information about differences for these routines, see subsection 3.7.2, page 3–28.

- Initial release: UNICOS 8.3

  Added a feature to set all blocks allocated by `malloc`(3) or freed by `free`(3) to the `indef` value. (This is a value that would cause an operand-range error (`ORE`) if used as an address, or a floating-point error (`FPE`) if used in a floating-point operation; see the `segldr`(1) man page for a complete definition.) Users can activate this feature by setting the `MEMINDEF` environment variable to a nonzero value, or by using the `M_INDEF` argument to the `mallopt`() function.

- Initial release: UNICOS 8.3

  Added a feature to cause a program to abort if an `sbreak`(2) call fails (that is, the program has run out of memory). This feature is activated by setting the `MEMABORT` environment variable to a nonzero value, or by using the `M_ABORT` argument to the `mallopt`(3) function.

- Initial release: UNICOS 8.3

  Added the following `mallopt()` commands: `M_BREAKSZ`, `M_MEMCHK`, `M_LOWFIT`, `M_INDEF`, and `M_ABORT`.

- Initial release: UNICOS 8.3

  Enhanced the memory manager tracing feature so that it traces calls to `brk`(2), `sbrk`(2), and `sbreak`(2) in addition to calls to the memory manager routines.

- Initial release: UNICOS 9.0

  The memory manager was changed to be more memory-efficient. This feature changes the memory manager's algorithm so that it uses less memory (a difference from 10% to a high factor, depending on the program), while the call is no more than 25% slower than the default algorithm. The new algorithm, called the `lowfit` algorithm, causes user programs to waste less memory, keeping total program size smaller.

  The reason this feature is needed is that, with certain memory allocation patterns, the heap can grow to a size much larger than the actual memory allocated by the user. The `lowfit` algorithm provides a method for decreasing memory usage while maintaining time efficiency.

  Users can activate this feature by setting the `MEMLOWFIT` environment variable to be `1` or by adding the following call to the beginning of a program:

```
mallopt(M_LOWFIT, 1);
```

For more information about the memory manager enhancements, see the `malloc`(3) man page.

2.8.5 *ER90 block mode support added*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

You can now write to an ER90 tape device in block mode from Fortran or with flexible file I/O (FFIO). Records written to this device must meet its restrictions on block sizes. For more information, see the `assign`(1) and `ffopen`(3) man pages.

Related publications

- *UNICOS Tape Subsystem User's Guide*, publication SG–2051

2.9 Operating system enhancements

The following subsections describe enhancements to the operating system for the UNICOS 9.0 release.

2.9.1 *UNICOS under UNICOS supported*

Users affected

Administrator, system analyst, operator

Supporting hardware

All Cray Research systems except CRAY J90 series and CRAY EL series

Initial release: UNICOS 8.0.3

This subsection briefly describes the UNICOS under UNICOS feature, which was initially supported in the UNICOS 8.0.3 release, and it also describes UNICOS 9.0 enhancements.

The UNICOS under UNICOS feature lets a site run two copies of the UNICOS operating system concurrently on one machine. The *host* UNICOS system boots normally with most of the system resources. Subsequently, a *guest* UNICOS system can be started by an authorized user. The UNICOS under UNICOS feature allows the guest system to be used for software testing and system upgrade, but not for production.

When you are planning to run or configure a guest system, remember the following:

- The UNICOS kernels of both the host and guest systems must contain support for the UNICOS under UNICOS feature.
- The guest system is a complete second operating system and requires the same minimum resources as any stand-alone kernel.

- Your IOS release level must support the higher release level of your guest or host system. For example, if UNICOS 9.0 is running as a guest under UNICOS 8.0.3, you need to run the IOS release level that supports UNICOS 9.0.

  For information about the IOS release level that supports the UNICOS 9.0 release, see subsection 7.1.2, page 7–2.

For information about using this feature to assist in upgrading your UNICOS system to a new release level, see the UNICOS under UNICOS articles in the *Cray Research Service Bulletin* (CRSB) and the *UNICOS under UNICOS Administrator's Guide*, publication SG–2156.

The host system performs the following tasks on behalf of the guest system:

- User exchanges
- Memory error handling
- Register parity error handling
- Physical I/O
- All SSD solid-state storage device and IOS request validation

The following safeguards are provided with this feature:

- Guest kernels can access only the memory allocated to them.
- Guest panics do not create panics on the host system.

With UNICOS 9.0, the following enhancements were made to the UNICOS under UNICOS feature, which was initially provided in the UNICOS 8.0.3 revision release. The features added in the UNICOS 9.0 release are available only when UNICOS 9.0 is used as the host system.

2.9.1.1 *RAM root file systems*

You can now specify that a RAM root file system be loaded in guest memory for use by the guest. The file specified can be either a `/dev/dsk` entry or a `dd`(1) copy. The `guest`(1) command makes cursory checks to ensure that the specified file contains an NC1 file system.

Additional guest parameter file changes are necessary to make use of the RAM file system. See *General UNICOS System Administration*, publication SG–2301, and the *UNICOS Installation Guide*, publication SG–2112, for more information about specifying and using RAM file systems.

Define the following parameter in your `guest.rc` file:

```
RAM_FILESYSTEMS=file
```

2.9.1.2 *Minimum host CPUs*

The administrator can now specify the minimum number of CPUs that are to remain in the host system at all times. Support for this specification has been added to the UNICOS under UNICOS menus in the UNICOS Installation/Configuration Menu System. Users with the `udb`(5) permbit `guestadmn` are also allowed to specify this number at guest boot time by using the `-N` *num* option to the `guest`(1) command. If there is no work to perform in the host, this number of CPUs will idle in the host, even if the guest requires CPU resources. Therefore, a setting of anything except 0 will not provide optimal CPU usage across both systems.

Define the following parameter in your `/usr/guest/Defaults` file:

```
MIN_HOST_CPUS=num
```

2.9.1.3 *Maximum guest CPUs*

The guest user can now request that the number of CPUs sent to the guest be fewer than the number configured in the host. The selection of CPUs begins with CPU 0. Only CPUs successfully started in the host will be selected.

Define the following parameter in your `guest.rc` file:

```
NUM_CPUS=num
```

2.9.1.4 *Guest memory allocation retry count*

On a busy system, the allocation of guest memory may require several incremental attempts. Previously, if the guest user's minimum memory requirements were not met after 10 attempts, the guest command would return any already allocated memory to the host and exit with an error. The user can now specify the number of memory retry attempts to make before failure. There is no limit.

Define the following parameter in your `guest.rc` file:

```
MEMORY_RETRIES=num
```

2.9.1.5 *General feature limitations*

The following are general limitations of this feature:

- The use of a CRAY T3D system from the guest is not supported.
- The guest system is not operated or monitored from the OWS.
- Tape configurations **must** be identical on the host and the guest. That is, all devices must be in the same position (ordinal) in the configuration. ER90 tapes, though unsupported in the guest, should still be used as place holders in the guest configuration.
- SSD space intended for the guest system must not be in use on the host (that is, SSD cannot be dynamically allocated at guest boot time).
- Multiple SSDs are not supported.
- Logical CPU 0 is required for booting a guest; therefore, it must not be down at guest boot time.
- Real-time applications may act differently on a guest or a host that has an active guest.
- Dedicating a CPU to a process (cpu(8)) in the guest or in a host with an active guest may result in sporadic execution of the process.
- If an alternate path is available for a disk device shared by the host and guest, both paths **must** be configured in each system.

For more information, see the new `guest`(1) man page.

Related publications

- *UNICOS under UNICOS Administrator's Guide*, publication SG–2156

2.9.2 *Generic data consolidator and front-end formatter added*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The system accounting generic consolidator and front-end formatter are more flexible versions of the `csacon`(8) and `csacrep`(8) commands. Together they allow you to do the following tasks:

- Consolidate the session file
- Consolidate multiple `pacct` files
- Generate a report or binary file based on the consolidated data

Both the Cray system accounting (CSA) and generic formatters are available. Consider the following factors when deciding which front-end formatter to use:

- The CSA front-end formatters require a source license, but the generic formatter does not.
- The generic front-end formatter output can be either ASCII or Cray binary data, where a binary number is always written as a 64-bit word. The CSA formatter can be modified to write 32-bit numeric values or extended binary coded decimal interchange code (EBCDIC) output.
- Both formatters process session record files created by `csabuild`(8). However, the generic formatter is also capable of processing multiple `pacct` files.

For more information, see the `csagcon`(8) and `csagfef`(8) man pages.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.3 *New accounting features*

Users affected

Administrator

Supporting hardware

All Cray Research systems

The features described in this subsection were added to the accounting software.

Initial release: UNICOS 8.3

- The system accounting feature retrieves, formats, and outputs more complete error messages using the UNICOS message system.
- A new accounting control feature has been added to the UNICOS 9.0 release. The accounting routines in the kernel and in Cray system accounting (CSA) have expanded the daemon accounting control to include record accounting control and command accounting control.
- The new `acctctl`(2) system call enables, disables, and checks the status for process, daemon, and record accounting. The `acctctl`(2) system call may be used instead of the `acct`(2) and `dacct`(2) system calls; it returns more information, including status information, in a more generic format.
- The `csaswitch`(8) command enables, disables, and checks the status of kernel, daemon, and record accounting.

  For more information, see the `acctctl`(2) and `csaswitch`(8) man pages.

Initial release: UNICOS 9.0:

- To improve I/O performance and to facilitate automatic data conversion, an I/O library layer was added to the accounting package. This enhancement also provides a simple way to change the underlying I/O method. For UNICOS 9.0, the underlying I/O method is changed from `fopen/fread/fwrite` to `ffopen/ffread/ffwrite` (FFIO).

  Use of the FFIO routines improved the overall performance of the accounting package; the system time was reduced by 50% to 75%.

  The size of the executable increased due to the additional use of the FFIO routines. The actual program size increased by the number of buffers used by the FFIO routines, which can be controlled by using the `assign`(1) command or a new environment variable (`ACCT_FFIO_OPT`).

For more information, see `ffopens` (on the `ffopen`(3) man page) and the *Application Programmer's I/O Guide*, publication SG–2168.

- To increase compatibility across release changes, all UNICOS 9.0 accounting tools are able to process accounting data generated on systems running UNICOS 8.0 or UNICOS 8.3. As needed, the data is converted automatically to UNICOS 9.0 format. Unlike previous releases, for UNICOS 9.0 you are no longer required to run the `csaconvert`(8) command to convert the prior accounting data to the current release format.

  However, if you access the prior accounting data on a regular basis, for performance reasons you should convert the data once using the `csaconvert`(8) command. This allows you to avoid the overhead of repeatedly converting the data automatically. In this instance, explicit conversion is preferred.

- The UNICOS Installation/Configuration Menu System can now be used when configuring accounting, thereby simplifying installation and upgrades.

  Accounting functionality remains the same, but the configuration file (`/etc/config/acct_config`), which contains configurable parameters used by the accounting commands, is updated by the tool.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.4 *X Window System-based interface added to manage user login accounts*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

An X Window System-based interface that uses the new `xadmin`(8) command provides a graphical user interface that has all the functionality of the `nu`(8) command. `xadmin` was created to make it easier for administrators to manage user login accounts. The X Window System-based interface is self-explanatory and requires no prior knowledge of the `nu` command. `xadmin` contains a tutorial for an overview of the command and context-sensitive help on specific topics.

For more information, see the `xadmin`(8) man page.

2.9.5 *User database (UDB) enhancements*

The user database (UDB) enhancements described in the following subsections are provided with the UNICOS 9.0 release.

2.9.5.1 *Clear-text password added to UDB*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

To improve ease of use, a `cpasswd` field was added to the user database (UDB). This clear text password is encrypted and stored in the user's record. The password content is not validated. This field is used on both UNICOS multilevel security (MLS) and non-MLS systems. On a UNICOS MLS system, only an appropriately authorized user can change this field.

For more information, see the `udbgen`(8) man page.

2.9.5.2 *Per-session sockbuf limit added by using UDB*

Users affected

All

Supporting hardware

All Cray Research systems

Inital release: UNICOS 8.0.4

Changes have been made to the UNICOS release since the UNICOS 8.0 major release. A new per-session `sockbuf` limit has been added to the UNICOS system. This limit is configurable on a per-user basis by using the standard UDB tools. An attempt to open a socket, accept a connection, or increase the per-sockbuf space limit is refused if the additional `sockbuf` space pushes the user's use over the session limit. This is intended to limit `sockbuf` use by any single user. Processes running with an effective UID of 0 are exempt.

For more information, see the `limit`(1), `accept`(2), `limit`(2), `getsockopt`(2), `netvar`(8), and `socket`(2) man pages.

Related publications

- *General UNICOS System Administration*, publication SG–2301
- *UNICOS Networking Facilities Administrator's Guide*, publication SG–2304
- Online revision of *UNICOS System Administration*, publication SG–2113 8.0, TCP/IP section

2.9.5.3 `jsocbflim` *supports new per-session* `sockbuf` *limit*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.4

The user database (UDB) includes the new `jsocbflim` field, which supports the new per–session `sockbuf` limit in UNICOS.

For more information about the new UDB field, see the `udbgen`(8) and `udblib`(3c) man pages.

2.9.5.4 *On-the-fly reset of maximum user ID in UDB*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The new `-m` option to the `udbgen`(8) command lets the system administrator reset the user database (UDB) maximum user ID (UID) value without having to rebuild the entire UDB from source.

The UDB maintains a record of the highest UID value assigned. This value is used to determine the UID when a new record is created by using `:uid:next:` syntax. Because this value is never decremented, it is possible to reach large UID values while leaving large ranges of UIDs unused. This option resets the UDB maximum UID value to the highest nondeleted UID.

For more information, see the `udbgen`(8) man page.

2.9.5.5 *Redesign of the UDB*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The user database (UDB) has been substantially rewritten to allow for future enhancements. The UDB now spans several files, and library routines for the UDB have been moved. However, the UDB interface remains substantially the same.

The following interface changes have been made to the UDB:

- The UDB has been extended. There are now extension files in a new subdirectory, `/etc/udb_2`, that physically separate public and private data in the UDB. (Private data includes encrypted passwords, security compartments, permission bits, etc.) Fields added appear in the extension files, with the exception of the user name, UID, group list, and account ID (`acid`) list.

Note: The files `/etc/udb` and `/etc/udb.public` continue to exist for compatibility; however, in a future release they will be removed, and all data will be moved into the new files.

- The `libudb`(3) library replaces the `udblib` library. The library source files have been moved from `libc/gen` to `libc/udb`.
- New fields have been added to support a minimum compartment set (`mincomps` field), shared memory and segments (`jshmsegs` and `jshmsize` fields), and dynamically expandable open file descriptors (`pfdlimit` field). The shared memory feature is supported on CRAY T90 systems only.

  For more information about the `mincomps` field, see subsection 4.2.11, page 4–8.
- The `limflags` field is no longer used, but is maintained for compatibility.
- New UDB permission bits (permbits) have been added to support retrieval of acid information (`askacid` permbit), group administration information (`groupadm` permbit), and the interprocess communication (IPC) persistence facility (`ipc-persist` permbit).
- There are two new library functions, `udb_strerror` and `udb_header_access`. The `udb_strerror` function translates a UDB error (`udb_errno` code) into a message string. The `udb_header_access` function is reserved for use in future releases.

For more information, see the `cll`(8), `getpermit`(2), `passwd`(5), `udb`(5), `udbgen`(8), `libudb`(3) (renamed from `udblib`(3)), and `udbsee`(1) man pages.

Related publications

- *General UNICOS System Administration*, publication SG–2301

2.9.5.6 *UDB fair-share scheduler flag holder enforcement added*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release contains a feature to enforce correct usage of the fair-share flags (the `shrflags` field) in the user database (UDB). If incorrect information in the UDB would cause the creation of an improperly labeled or positioned lnode, the system issues an error, and the user is prevented from logging in or submitting jobs.

The new `shrtree`(8) command can be used to verify the fair-share hierarchy in the UDB. This command produces messages that indicate possible errors.

The major impact of this feature will be felt by sites that run the fair–share scheduler with Share by Account mode enabled. The `SHAREHOLDER` flag must be set in the `shflags` field for UDB records that define account IDs (acids).

For more information, see the `shrtree`(8) and `shradmin`(8) man pages.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.6 *Unified Resource Manager (URM) enhancements*

The Unified Resource Manager (URM) enhancements described in the following subsections are provided in the UNICOS 9.0 release.

2.9.6.1 *URM checkpointing*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

The URM checkpointing enhancements described in this subsection were added to UNICOS.

- Initial release: UNICOS 8.3

  The URM checkpoint feature improves user recoverability by allowing the user to periodically checkpoint an active session. This protects against lost work due to an uncontrolled system shutdown and allows an interactive session to be checkpointed as part of a controlled system shutdown.

The URM configuration parameter set by the system administrator selects whether checkpointing is done only at shutdown, or periodically and at shutdown.

To determine whether a user has any checkpointed sessions, query URM by using the new `restart` subcommand to `rmgr`(1) as follows:

```
rmgr-> view restart username
```

This command returns a numbered list of checkpointed sessions.

To restart a checkpointed session, invoke the `restart` subcommand as follows:

```
rmgr-> restart username chkpnt#
```

- Initial release: UNICOS 8.0.3/8.3

  URM now provides a boost for checkpointed jobs. This boost is added to the URM job rank to provide preferential treatment for jobs that must be restarted. A new selection in the UNICOS Installation/Configuration Menu System allows administrators to change the default boost value.

For more information, see the `chkptint`(1) and `rmgr`(1) man pages.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.6.2 *New URM user job display command*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The new `ustat`(1) command displays session (job) information from the URM, including jobs queued by the Network Queuing System (NQS). This command provides a user-friendly interface to the URM system load information in `rmgr`(1). The display can be organized by job status, session initiator, requested resources, or other categories.

By default, only the super user and authorized administrators can view information for all jobs. However, the system administrator can enable an optional control to let all users on the system view all jobs.

For more information, see the ustat(1) man page.

2.9.6.3 *URM capability to change minimum rank of NQS job priority*

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The URM now has the capability to change the minimum rank of a batch job (called the *job priority* in the Network Queuing System (NQS)). As a result, for systems that use both NQS and URM, the functionality of the following qmgr(8) subcommand has been restored:

```
modify request requestid priority_limit = priority
```

By default, NQS sets the minimum rank of jobs in the job backlog. The new usetjob(8) command lets the URM administrator change the minimum rank of a specific job or jobs to affect the URM rank of those jobs.

The ustat(1) command has a new option, -m, to display the minimum rank of a job. The rmgr(1) directive view can also be used to display minimum rank values.

For more information, see the rmgr(1), usetjob(8), and ustat(1) man pages.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.6.4 *URM smoothing algorithm enhanced*

Users affected

System analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The URM smoothing algorithm was enhanced to smooth only when resource use drops and to show increases in resource consumption immediately.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.6.5 *User exit enhancements to URM*

Users affected

System analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The URM user exit was enhanced to allow URM to manage site-specific information attached to a job. If a user exit creates job-specific information, the URM maintains a pointer to the memory allocated for this information and automatically frees memory when necessary.

To access this capability, the user exit must use the new `uxit` union in the Job data structure as a pointer and set the new `JOBHIST_UXIT` flag in the history field.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.6.6 *Fair-ratio share policy added to URM*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The behavior of the Network Queuing System (NQS) when the URM is enabled has been improved. In previous releases, enabling URM changed the scheduling behavior of NQS slightly.

URM now contains an additional scheduling policy, the fair-ratio share policy, to allow NQS to schedule jobs as expected (that is, as when URM is not enabled).

Administrators can enable this optional policy by setting `Share policy` to `fair_ratio` in the UNICOS Installation/Configuration Menu System. Once the policy is enabled, usage values are calculated by including the usage of the subject node, proportional to the sum of the usage in each level of the fair-share hierarchy. The default policy is `Standard`, which calculates usage based only on the usage of the terminal node in the hierarchy.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.6.7 `rmgr(1)` *command accepts lowercase subcommands*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The `rmgr(1)` command now accepts lowercase commands (such as `view`). The previous form of commands (for example, `View`) is still valid.

For more information, see the `rmgr(1)` man page.

2.9.6.8 *URM capability to move log file directory*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.4

The URM now has the capability to change the default directory used to store log files. As a result, system administrators have more flexibility in the storage of potentially large amounts of information.

The URM daemon, `urmd(8)`, contains the new `-l` option to accomplish this function. In addition, the `rmgr(1)` command contains the new directive `Set Log` directory, which accomplishes this function when the URM is executing.

For more information, see the `rmgr(1)` and `urmd(8)` man pages.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.6.9 *New* `urmsnap` *command for URM*

Users affected

Administrator

Supporting hardware

All Cray Research systems that support UNICOS 8.3/9.0

Initial release: UNICOS 8.3

The new `urmsnap(8)` command captures the current URM configuration and displays it in a readable format. This command allows administrators to examine the current URM configuration when making changes to URM parameters.

For more information, see the `urmsnap(8)` man page.

2.9.7 *Fair-share scheduler enhancements*

The fair-share scheduler enhancements described in the following subsections are provided with the UNICOS 9.0 release.

2.9.7.1 *User-level fair-share scheduler mode added*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The user-level fair-share scheduler feature provides a user exit and duplicates kernel functionality at the user level. The general interface to fair-share remains largely the same, with the following enhancements:

- The fair-share scheduler's calculations can be performed by the fair-share daemon (`shrdaemon`(8)) instead of the kernel. In this user-level fair-share mode, `shrdaemon` replaces the kernel functions that apply the scheduling policy algorithms. This optional mode is enabled by using the `USRLEVLFSS` flag in `shradmin`(8).

  Note: The user-level fair-share mode is optional in the UNICOS 9.0 release. It will be the default in future releases of the UNICOS system, when the kernel will no longer have the capability to apply the scheduling policy algorithms.

- The user exit feature allows sites to customize the CPU scheduling policy. The user exit is available only when the `USRLEVLFSS` flag is enabled in `shradmin`(8).

- The new `policy`(2) system call provides an interface to the kernel for the fair-share daemon and allows access to the fair-share constants structure, `sh_consts`. This system call allows site selection of other CPU allocation policies to be added in future releases.

- The functionality of the `limits`(2) system call is now limited to lnode operations. The `policy`(2) system call handles other fair-share functionality.

Note: The `L_GETCOSTS` and `L_SETCOSTS` functions of the `limits`(2) system call have been removed in the UNICOS 9.0 release. The `policy`(2) system call provides the `GET_COSTS` and `SET_COSTS` actions as a replacement for the `L_GETCOSTS` and `L_SETCOSTS` functions.

For more information, see the `shrdaemon`(8), `shradmin`(8), `policy`(2), and `limits`(2) man pages.

2.9.7.2 *Fair-share* `shradmin`(8) *command functionality changes*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The `shradmin`(8) command supports the following new fair-share functionality:

- The new `-c` option sets the maximum number of memory clicks that can be used by the aggregate of user or account processes attached to an lnode.
- The new `-n` option specifies the interval used for updating lnode information in the user database (UDB).
- The new `-p` option sets the maximum number of processes allowed per user or account lnode.
- The `-t` option now reports charges for terminal I/O operations.

For more information about these new options, see the `shradmin`(8) man page.

For more information about changes to the fair-share interface, see subsection 3.8.3, page 3–30.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.7.3 *CPU quotas synchronization added*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The CPU quota synchronization feature allows the administrator to synchronize the CPU quota-used field of the user database (UDB) with the corresponding field in the user's active lnode.

This capability is necessary to successfully alter a user's CPU quota (for example, when clearing the CPU quota-used field in the UDB). Without this synchronization feature the administrator cannot update quotas while the system is in multiuser mode, because the fair-share scheduler daemon overwrites any changes made to the UDB for active users.

The fair-share command `shrsync`(8) is used to synchronize changes in the fair-share information. The `-q` option has been added to synchronize the CPU quota-used information.

For more information, see the `shrsync`(8) man page.

Note: The CPU quota value itself is **not** synchronized. This value is not explicitly stored in the lnode, but rather is used in combination with the user's CPU limits and placed in the user's lnode.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.7.4 `crash` *command displays* `shrconst` *structure*

Users affected

Administrators

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The `crash` command supports debugging the fair–share scheduler with the new `shrc` subcommand, which displays the shrconst structure.

For more information, see the `crash`(8) man page.

2.9.8 *System activity monitoring enhancements*

The system activity monitoring enhancements described in the following subsections are provided with the UNICOS 9.0 release.

2.9.8.1 `sam` *enhanced*

Users affected

Administrator, end user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The functionality of the system activity monitoring command, `sam`(8), has been improved to report shared-text segments in the memory map display and swapped-out secondary data segments in the swap map display.

For more information, see the `sam`(8), `csam`(8), and `xsam`(8) man pages.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.8.2 `sar` *and* `tsar` *enhanced*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The system activity reporting commands, `sar`(1) and `tsar`(8), have been enhanced to track and report per-CPU idle time that resulted from processes waiting for I/O to complete. Three new options have been added to the `sar`(1) user command to display the following information:

- Network File System (NFS) server and client activity. (The `-r` option reports output from the `nfsstat`(8) command.)
- Network packet traffic. (The `-n` option reports output from the `netstatd`(8) command.)
- Summary format. (The `-S` option reports raw totals from the last UNIX restart or from a specified interval.)

The following changes have been made to the `sar`(1) command:

- The display of the `-X` option has been altered to show the total abnormal exchanges, rather than a rate that is almost always 0.
- `ldcache` statistics are reset to 0 upon reconfiguration of the system `ldcache` parameters.

For more information, see the `sar`(1) and `tsar`(8) man pages.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.8.3 `diskusg` *command rewritten*

Users affected

Administrator

Supporting hardware

All Cray Research systems except CRAY J90 series and CRAY EL series

Initial release: UNICOS 8.3

The `diskusg`(8) command was rewritten to improve performance by changing the underlying data collection algorithm. Also, it provides site exits that allow you to process data for all inodes or for all file inodes, and to generate a customized report. `diskusg` reads and processes an entire NC1 file system inode region on a single request.

No changes have been made to the `diskusg` command line; accounting scripts will work without changes. No changes are required to use the new `diskusg` command; however, the new algorithm will require more memory because it reads the entire disk inode region into memory rather than accessing each inode with a separate request. `diskusg` resides in a separate subdirectory within the accounting directory structure.

For more information, see the `diskusg`(8) man page.

Related publications

- *UNICOS Resource Administration*, publication SG–2302

2.9.9 `nschedv` *command supports kernel scheduler enhancement*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.3/8.3

The UNICOS kernel scheduler now includes the `small_proc` interactive performance enhancement, which lets an administrator set a size threshold for small processes. Interactive processes below this threshold will be held in memory while there is interaction from a terminal within a specified interactive interval.

Two new parameters were added to nschedv(8) to support this change: the -y option sets the size limit of a small process, and the -Y option specifies the interactive interval.

For more information, see the nschedv(8) man page.

2.9.10 *New options for* dump *and* restore *commands*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

New options were added for the dump(8) and restore(8) commands.

The dump(8) command supports the new -B option, which allows an administrator to specify the number of 32678-byte blocks that are output in a single listio(2) operation. If you are not piping your output, this enables you to tune your output according to the specific needs of your application. The default value of the option is 1 for pipes, which is the only value that pipes support, and 20 for other output files.

The dump(8) command also supports the new -Z option, which causes the dump(8) command to skip all regular nonmigrated files larger than the size you specify. An administrator can use this option to specify that the backup of a migrated file system contain no files larger than the minimum migrated size.

The restore(8) command supports the -T option, which generates a trailing file name list that contains the names of all files that may have changed while dump(8) was running.

For more information, see the dump(8) and restore(8) man pages.

2.9.11 crash *command enhancements*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.3 / 8.3

The crash(8) administrator command has been enhanced to display information about the following features.

- DCE Distributed File Server (DFS) data structures

  The following subcommands were added to support examination and debugging of DCE DFS structures:

  `aggr`, `ccall`, `cct`, `ch`, `chtable`, `cm_conn`, `cm_serv`, `dcache`, `dfsmisc`, `dfsstat`, `fid`, `fshost`, `scache`, `sct`, `tkc`, `tkm`, `tkset`, `tpq`, `volume`

- Fair–share scheduler

  The new `shrc` subcommand displays the fair–share `shrconst` structure.

  For information on this subcommand, see the `crash`(8) man page. For more information, see subsection 2.9.7.4, page 2–63.

- IPI packet driver traces

  The subcommands `pktdi` and `pktdt` were added to support formatting packet driver traces for IPI–3 attached disk and tape storage devices.

  For information on these subcommands, see the `crash`(8) man page. For more information on IPI–3 support, see subsection 2.3.9, page 2–14.

- NFS and RPC data structures

  The `crash` subcommands `svc_data` and `svc_xprt` were added to support formatting NFS and RPC data structures.

  For information on these subcommands, see the `crash`(8) man page. For more information on NFS support, see section 2.5.15, page 2–28.

- Shared memory (Cray T90 series systems only)

  The new `shm` subcommand displays information on shared memory segments. In addition, the `mem`, `job`, and `sess` subcommands have been changed to display shared memory segments.

  For information on these subcommands, see the `crash`(8) man page. For more information on the shared memory feature, see subsection 2.1.3, page 2–4.

- System V semaphores and message queues

  The new subcommands `msgq`, `sema`, and `semundo` display information on semaphore and message queue data structures. In addition, the `job` and `sess` subcommands have been changed to display semaphores and messages.

  For information on these subcommands, see the crash(8) man page. For more information on semaphores and message queues, see subsection 2.9.25, page 2–78.

- UNICOS under UNICOS

  The following subcommands were added to support examination and debugging of a guest UNICOS system:

  `gcom`, `gch`, `gcx`, `grt`, `gpf`, `gpi`, `gqp`, `gsn`, `guest`

  For information on these subcommands, see the `crash`(8) man page. For more information on the UNICOS under UNICOS feature, see subsection 2.9.1, page 2–46.

In addition, the documentation for `crash` has been enhanced to include information on subcommands and options that were formerly available only with the `help` subcommand in `crash`. For more information, see the `crash`(8) man page.

2.9.12 *Dynamic control of disk recovered errors*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

An administrator can stop the routing of recovered disk error messages to the console by executing the `pddconf`(8) command and specifying `racerroff` as the *function* parameter. Specifying `racerron` restores the routing of the messages to the console (the default state).

For more information, see the `pddconf`(8) man page.

2.9.13 *Ability to disable write-behind to a mirrored device*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

An administrator can disable write-behind on write I/O to a mirrored disk device by using the new `-w` option to the `mddconf`(8) command. Disabling write-behind guarantees that the data gets to a device. However, using this feature does result in slower performance.

For more information, see the `mddconf`(8) man page.

Related publications

- *General UNICOS System Administration*, publication SG–2301

2.9.14 *New* `hddmon` *command*

Users affected

End user, administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.4

The `hddmon`(8) command has been added to the UNICOS 9.0 release. The `hddmon`(8) command allows an adminstrator or operator to monitor and control HIPPI disk array devices.

For more information, see the `hddmon`(8), `hdd`(4), and `mnu`(4) man pages.

2.9.15 *Using a dump partition as a swap device supported*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

You can use the same physical slice for both a dump partition and a swap partition. Because the dump partition is in use only at dump time and remains idle while the system is running, it can be used as one of the partitions of the swap device. Nothing is saved from the swap device between system boots, which allows you to put a dump image on top of the swap data.

When you reboot the system, if there is a swap partition being shared with the dump device, and a dump currently resides on that partition, the `cpdmp`(8) command moves the dump to the file system.

Related publications

- *General UNICOS System Administration*, publication SG–2301

2.9.16 *Tape subsystem enhancements*

The tape subsystem enhancements described in the following subsections are provided with the UNICOS 9.0 release.

For information about an enhancement to the `tpmnt`(1) command, see subsection 2.6.13, page 2–40.

2.9.16.1 *Character-special tape interface support added*

Users affected

End user, programmer, administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.0.3/8.3

The addition of the character-special tape interface to the UNICOS tape subsystem creates another method of accessing tape devices.

This new character-special tape interface method is similar to the traditional UNIX method of accessing tape devices; you have unstructured access to the tape devices. That is, to manage your tapes you can use standard UNIX commands and `ioctl`(2) requests. You may find this access method particularly useful for porting tapes from one UNIX system to another.

In previous releases, tape devices could be accessed only by using a kernel device driver and the tape daemon. The tape daemon assists the device driver by performing additional functions such as tape resource management, device management, volume mounts and dismounts through operator communication or autoloader requests, label processing, volume switching, and error recovery. The new character-special tape interface does not support these capabilities.

For more information, see the mt(1B), close(2), open(2), tape(4), and tpdaemon(8) man pages.

Related publications

- *UNICOS Tape Subsystem Administrator's Guide*, publication SG–2307
- *UNICOS Tape Subsystem User's Guide*, publication SG–2051

2.9.16.2 *Unstructured access to tape hardware supported*

Users affected

Administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

New options to the tpformat(8) and tplabel(8) commands allow administrators to bypass the tape daemon and have unstructured access to the tape hardware to label and format tapes. The new -C option specifies that the character-special tape interface (instead of the tape daemon) be used to perform the operation. The -D option specifies the device to which the request is issued.

For more information, see the tpformat(8) and tplabel(8) man pages.

Related publications

- *UNICOS Tape Subsystem Administrator's Guide*, publication SG–2307
- *UNICOS Tape Subsystem User's Guide*, publication SG–2051

2.9.16.3 *Dynamic device group reassignment supported*

Users affected

Administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The dynamic device group feature allows an operator or administrator to reassign down devices to a different device group than the group in which they were originally configured. Sites can use the tpconfig(8) command to move devices from one device group to another depending on load and requirements.

This feature is particularly useful for performing backups when an operator wants to temporarily dedicate a set of devices for the backup operation. Once the backups are complete, the operator may configure the devices back into the public device group for normal use.

For more information, see the `tpconfig`(8) man page.

2.9.17 `tplist` *utility supports Cray/REELlibrarian*

Users affected

Programmer, administrator, end user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The `tplist`(1) command has been enhanced so that it supports the Cray/REELlibrarian (CRL) product and the associated `tpmnt`(1) options. You can now specify additional `tplist` parameters for the primary and secondary `tpmnt`(1) commands. You can also set the `no unload` flag to keep the primary volume mounted.

Administrators can make CRL and front-end servicing (FES) on IBM MVS systems available at the same time. If both are set to `on`, and the related mandatory flags are not set, users may select which one to use.

For more information, see the `tplist`(1) man page.

2.9.18 *IPI support enhancements*

The Intelligent Peripheral Interface (IPI) support enhancements described in the following subsections are provided with the UNICOS 9.0 release.

2.9.18.1 `ipi3_clear` *command options added*

Users affected

Administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The `ipi3_clear`(8) command has two new options. Administrators can use the new `-c` option to terminate all outstanding activity on a channel and the new `-r` option to issue a slave reset as part of the clear processing.

For more information, see the `ipi3_clear`(8) man page.

2.9.18.2 *IOP-level control of IPI-3 and IPI packet drivers added*

Users affected

Administrator, operator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Administrators can now use the `ipi3_start`(8) and `ipi3_stop`(8) commands to stop and restart an IPI-3/IPI subsystem or a single I/O processor (IOP) within the subsystem. These commands also allow an IOP restart or shutdown to be initiated from an interrupt that was originated by the `inform`(8) command.

For more information, see the `ipi3_start`(8) and `ipi3_stop`(8) man pages.

2.9.19 *Dynamic allocation of file descriptors supported*

Users affected

Administrator, programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

In the past, processes running under the UNICOS system could open no more than 64 files simultaneously. This restriction has proven unacceptable for many programs including database applications and, more recently, applications for the CRAY T3D system.

This restriction was alleviated somewhat in UNICOS 8.0 by a feature that increased the maximum open file limit to 255 files.

The UNICOS 9.0 release implements the second phase of dynamic open file limits. This feature allows the administrator to control the system limit on open files. This system limit is set by using the new `_sc_cray_open_max` variable in `sysconf`(2). The default limit at release is 16,384 files.

Because open files consume other limited system resources (such as inode and file table entries), use of this feature can be controlled by the new `pfdlimit` field in the user database (UDB).

For more information, see the `udbgen`(8), `libudb`(3), `nu`(8), `xadmin`(8), `sysconf`(2), and `limit`(2) man pages.

For information about programming issues related to this feature, see subsection 3.8.4, page 3–31.

2.9.20 *File system enhancements*

File system enhancements described in the following subsections are provided with the UNICOS 9.0 release.

2.9.20.1 *Default number of primary partitions in a file system set to 4*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

To improve file system performance, an upper limit on the number of primary partitions in a file system that are maintained with current data has been set to 4. Other primary partitions continue to contain primary data, but do not have current copies of the file system dynamic block or block allocation bit maps. This feature takes effect at file system mount time.

To further reduce the number of primary partitions updated, the `mkfs`(8) command has been changed to set the number of primary partitions to 4 whenever the number of partitions in a file system is 8 or greater. The remaining partitions are designated as secondary. This is the default behavior of `mkfs`(8). You can still configure a file system with any number of primary partitions by specifying the `-p` option to `mkfs`(8).

Setting a maximum number of updated primary partitions should increase the file system performance of file systems with a large number of primary partitions. This performance increase occurs because the overhead involved in updating multiple primary partitions is reduced.

The `fsck`(8) command has been modified to account for this change. File systems created with previous releases of the UNICOS operating system will encounter `fsck`(8) errors and may require corrective action. For information about how to correct this situation, see subsection 3.8.5, page 3–32.

For more information, see the `fsck`(8) and `mkfs`(8) man pages.

2.9.20.2 *File system scrub added*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The file system scrub feature allows disk blocks to be scrubbed (zeroed) on a file system basis. This capability can be enabled at the time the file system is created by using the `-e` option of the `mkfs`(8) command, or it can be toggled by using the `-e` option of the `setfs`(8) command.

For more information, see the `mkfs`(8) and `setfs`(8) man pages.

2.9.21 *UNICOS Shared File System (SFS) support*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The UNICOS shared file system (SFS) allows multiple Cray Research systems to access one or more shared file systems in a read/write fashion. Shared file systems provide most of the features present in the NC1 file system, such as device striping, mirrored file systems, and asynchronous I/O.

To the user, a shared file system looks no different than an NC1 file system, except that the user can access the file system from any machine in an SFS cluster that is configured to use that shared file system. Systems in an SFS Cluster can mount any of the shared file systems configured on the shared media acessible to the machines in that cluster.

Shared file systems work only with a HIPPI-based disk subsystem, such as the ND-40 network disk device. The UNICOS SFS feature is not efficient for shring small files, or for files on which small I/O operations are frequent. Instead, the UNICOS SFS feature is designed for application which can do large I/O operations.

The *UNICOS Installation/Configuration Menu System* supports the configuration of shared file systems using the special `configsfs` program rather than the standard `install` program.

The `esdmon`(8), `sfsd`(8), `smpmon`(8), and `sfs`(4) man pages describe commands and features that support the SFS feature. The `mkfs`(8) and `setfs`(8) man pages have been updated to support shared file systems.

To use the UNICOS SFS feature you must obtain a software license. For information about licensing this product, see subsection 7.2.3.3, page 7–9 or contact your Cray Research representative.

Related publications

- *UNICOS Shared File System (SFS) Administrator's Guide*, publication SG–2114

2.9.22 *Excess user errors controlled*

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release maintains a new table that counts error exits, operand-range errors, and program-range errors for every program in the system, and allows the administrator to set maximum allowable values for any of these errors.

This feature can prevent situations, for example, where a user program overwrites itself and generates a series of errors that cause the kernel to loop by sending a `SIGERR` signal to a program that has turned off signal processing. This process would accumulate system time until the program reached its CPU time limit.

This feature is disabled by default. By means of the installation tool, you can set the maximum levels for the errors that the system counts. These maximums are set through the following parameters: `MAXUSRPRE`, `MAXUSRORE`, and `MAXUSRERR`. You can change those values at run time by using the new `seterr`(8) command.

For more information, see the `seterr`(8) and `cpu`(4) man pages.

Related publications

- *UNICOS Configuration Administrator's Guide*, publication SG–2303

2.9.23 *Dynamic allocation of the* NPBUF *parameter added*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

An administrator can allocate the number of physical I/O headers (pbufs) at boot time by modifying the NPBUF configuration parameter.

Related publications

- *UNICOS Configuration Administrator's Guide*, publication SG–2303

2.9.24 *Ill-formed I/O split into physical and buffered chunks*

Users affected

Administrator, programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The ill-formed I/O splitting feature improves I/O performance by providing a mechanism to automatically use raw I/O rather than system buffer cache for certain I/O requests. There are two general kinds of I/O that are affected.

The first kind of I/O is on files opened with the O_RAW flag. Large requests now can avoid using the system buffer cache to the extent possible. Previously, if any portion of a request to the I/O device was not well-formed, the entire request was routed through the buffer cache. In UNICOS 9.0, requests that are not well-formed are broken up into well-formed and non-well-formed parts. The well-formed parts use raw I/O, and parts that are not well-formed use the system buffer cache. A threshold is used to limit this behavior to requests that are sufficiently large.

The second kind of I/O to derive benefit from this feature is I/O done on files where the O_RAW flag was not specified when the file was opened, but where request sizes are over a certain size. I/O requests over this minimum size are handled in the same way as large raw requests.

Performance is improved on codes that do large granularity I/O that is well-formed for disks with small sectors but is not well-formed for disks with larger sectors. Performance is also improved where applications do large granularity I/O that currently passes through the system buffer cache but does not fit entirely within the cache.

The criteria used for how large a request must be before being automatically split is controlled by the `nc1_min_raw` system variable in the `config.h` file. The minimum request size is determined by multiplying `nc1_min_raw` times the device I/O unit. Any request which equals or exceeds this value will be automatically split.

In very rare cases, some degradations are possible with this feature. For large granularity I/O to files that fit in the cache, where other I/O activity is very light, this feature will force I/O to disk where it previously might have resided entirely in buffer cache. If degradations are seen and are deemed to be unacceptable, the threshold can be changed to disable the feature.

You can disable ill–formed I/O splitting by setting the `nc1_min_raw` system variable in the `config.h` file to zero. The default value for `nc1_min_raw`, defined as `NC1MINRAW` in the `config.h` file, is currently set to 20.

2.9.25 *IPC semaphores and message queues supported*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

Support was added for the UNIX System V interprocessor communication (IPC) mechanism. This mechanism introduces three named object types to the UNICOS system: shared memory segments (CRAY T90 series only); semaphores; and message queues.

For information about shared memory segments (CRAY T90 series only), see subsection 2.1.3, page 2–4.

The `ftok`(3) library routine (described on the `stdipc`(3) man page), used in conjunction with the IPC system calls, generates IPC facilities. The following new system calls support use of semaphores and message queues:

System call	Description
`msgctl`(2)	Provides message control operations
`msgget`(2)	Accesses the message queue
`msgrcv`(2)	Reads a message from a message queue
`msgsnd`(2)	Sends a message to a message queue
`semctl`(2)	Provides semaphore control operations

System call	Description
`semget`(2)	Provides access to semaphore identifiers
`semop`(2)	Provides general semaphore operations

The following configuration parameters have been added to the UNICOS Installation/Configuration Menu System in support of IPC semaphores: `SEMNI`, `SEMMNS`, `SEMMNS`, `SEMMNU`, `SEMMSL`, `SEMOPM`, `SEMUME`, `SEMVMX`, and `SEMAEM`.

The following configuration parameters have been added to the UNICOS Installation/Configuration Menu System in support of message queues: `MSGMAX`, `MSGMNB`, `MSGMNI`, `MSGSSZ`, `MSGTQL`, and `MSGSEG`.

Changes were made to the UNICOS multilevel security (MLS) feature to support IPC semaphores and message queues; for more information, see subsection 4.2.8, page 4–5.

The new `crash`(8) subcommands `msgq`, `sema`, and `semundo` display information on semaphores and message queues.

For more information on this feature, see the following man pages: `ipcrm`(1), `ipcs`(1), `msgctl`(2), `msgget`(2), `msgrcv`(2), `msgsnd`(2), `semctl`(2), `semget`(2), `semop`(2), `stdipc`(3), `ipc`(5), `msg`(5), `sem`(5), `ipc`(7), and `crash`(8).

Related publications

- *UNICOS Configuration Administrator's Guide*, publication SG–2303

2.10 XPG4 compliance

The following subsections describe additions to the UNICOS 9.0 release for compliance with the X/Open XPG4 standard. The XPG4 standard supports all POSIX functionality.

For more information about XPG4 compatibility issues, see subsection 3.9, page 3–36.

2.10.1 *XPG4 compatibility added*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

To comply with the XPG4 standard, some existing functions have been changed and other functions mandated by the standard have been added. No functions or capabilities have been removed. However, include file changes may require user code changes.

For more information about XPG4 compatibility issues, see subsection 3.9, page 3–36.

For more information, see the `catgetmsg`(3), `catgets`(3), `catmsgfmt`(3), `catopen`(3), `regexp.h`(3), `lsearch`(3), `tsearch`(3), `getc`(3), `putc`(3), `getpass`(3), `swab`(3), `chroot`(2), and `ulimit`(2) man pages.

2.10.2 *Utilities now use XPG4-conformant message system*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

UNICOS user-level utilities now issue their messages using the XPG4-conformant UNICOS message system.

The new messages issued by user-level utilities are changed to the default utility message format of **group-number: command message**. For more information about message formatting, see the `explain`(1) man page.

2.10.3 *Locale definition and library routines supported*

Users affected

Administrator, end user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The `localedef`(1) command has full POSIX 1003.2 and XPG4 functionality and issues its messages using the UNICOS message system.

This feature allows users (for themselves) and administrators (for entire systems) to define the sets of culture conventions supported by commands and library routines. This includes information about languages, code sets, monetary formats, numeric formats, and time/date formats. Memory use increases significantly only if large collation tables will be used.

For more information, see the `localedef`(1), `setlocale`(3), `string`(3), `localeconv`(3), `nl_langinfo`(3), and `strfmon`(3), man pages.

Related publications

- *General UNICOS System Administration*, publication SG–2301

2.10.4 *XPG4 standard I/O internationalization supported*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

UNICOS 9.0 internationalizes I/O usage and adds XPG4 support by implementing basic existing I/O functions to convert input from multibyte characters to wide characters and to convert output from wide to multibyte.

For more information, see the `mbstring`(3), `strftime`(3), `strtod`(3), `strtol`(3), and `wstring`(3) man pages.

2.10.5 *User commands in compliance with XPG4*

Users affected

End user

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The following user commands functionally comply with the XPG4 standard.

For compatibility issues related to commands, see subsection 3.9.3, page 3–38, and subsection 3.9.4, page 3–39.

- `admin`(1)
- `alias`(1)
- `ar`(1)
- `asa`(1)
- `at`(1)

- awk(1)
- basename(1)
- batch(1)
- bc(1)
- bg(1)
- c89(1)
- cal(1)
- cat(1)
- cd(1)
- cflow(1)
- chgrp(1)
- chmod(1)
- cksum(1)
- cmp(1)
- comm(1)
- command(1)
- compress(1)
- cp(1)
- crontab(1)
- csplit(1)
- ctags(1)
- cut(1)
- cxref(1)
- date(1)
- dd(1)
- delta(1)
- df(1)
- diff(1)
- dirname(1)
- du(1)
- echo(1)
- ed(1)
- egrep(1)
- env(1)
- ex(1)
- expand(1)

- expr(1)
- false(1)
- fc(1)
- fg(1)
- fgrep(1)
- file(1)
- find(1)
- fold(1)
- fort77(1)
- gencat(1)
- get(1)
- getconf(1)
- getopts(1)
- grep(1)
- hash(1)
- head(1)
- iconv(1)
- id(1)
- jobs(1)
- join(1)
- kill(1)
- ksh(1)
- lex(1)
- ln(1)
- locale(1)
- localedef(1)
- logger(1)
- logname(1)
- lp(1)
- ls(1)
- m4(1)
- mailx(1)
- make(1)
- man(1)
- mesg(1)
- mkdir(1)

- `mkfifo`(1)
- `more`(1)
- `mv`(1)
- `newgrp`(1)
- `nice`(1)
- `nl`(1)
- `nm`(1); incompatibility introduced with the UNICOS 8.3 release
- `nohup`(1)
- `od`(1); incompatibility introduced with the UNICOS 8.3 release
- `paste`(1)
- `patch`(1)
- `pathchk`(1)
- `pax`(1)
- `pr`(1); incompatibility introduced with the UNICOS 8.3 release
- `printf`(1)
- `prs`(1)
- `ps`(1)
- `pwd`(1)
- `read`(1)
- `renice`(1)
- `rm`(1)
- `rmdel`(1)
- `rmdir`(1)
- `sact`(1)
- `sccs`(1)
- `sed`(1)
- `sh`(1)
- `sleep`(1)
- `sort`(1)
- `split`(1)
- `strings`(1)
- `strip`(1)
- `stty`(1)
- `tabs`(1)
- `tail`(1)
- `talk`(1)

- tee(1)
- test(1)
- time(1)
- touch(1)
- tput(1)
- tr(1)
- true(1)
- tsort(1)
- tty(1)
- type(1)
- ulimit(1)
- umask(1)
- unalias(1)
- uname(1)
- uncompress(1)
- unexpand(1)
- unget(1)
- uniq(1)
- uucp(1)
- uudecode(1)
- uuencode(1)
- uustat(1)
- uux(1)
- val(1)
- vi(1)
- wait(1)
- wc(1)
- what(1)
- who(1)
- write(1)
- xargs(1)
- yacc(1)
- zcat(1)

2.10.6 *System calls in compliance with XPG4*

Users affected

End user, programmer, system analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The following system calls functionally comply with the XPG4 standard:

- `access`(2)
- `alarm`(2)
- `chdir`(2)
- `chmod`(2)
- `chown`(2)
- `close`(2)
- `creat`(2)
- `dup`(2)
- `exec`(2)
- `exit`(2)
- `fcntl`(2)
- `fork`(2)
- `fpathconf`(2)
- `fstat`(2)
- `fsync`(2)
- `geteuid`(2)
- `getgid`(2)
- `getgroups`(2)
- `getpgrp`(2)
- `getpid`(2)
- `getppid`(2)
- `getuid`(2)
- `kill`(2)
- `link`(2)
- `lseek`(2)
- `mkdir`(2)
- `mkfifo`(2)
- `msgctl`(2)
- `msgget`(2)
- `msgrcv`(2)
- `msgsnd`(2)

- nice(2)
- open(2)
- pathconf(2)
- pause(2)
- pipe(2)
- read(2)
- rename(2)
- rmdir(2)
- semctl(2)
- semget(2)
- semop(2)
- setgid(2)
- setpgid(2)
- setsid(2)
- setuid(2)
- shmat(2); CRAY T90 only
- shmctl(2); CRAY T90 only
- shmdt(2); CRAY T90 only
- shmget(2); CRAY T90 only
- sigaction(2)
- signal(2)
- sigpending(2)
- sigprocmask(2)
- sigsuspend(2)
- stat(2)
- sysconf(2)
- tcgetpgrp(2)
- tcsetpgrp(2)
- time(2)
- times(2)
- ulimit(2)
- umask(2)
- uname(2)
- unlink(2)
- utime(2)

- wait(2)
- write(2)

2.10.7 *Library routines in compliance with XPG4*

Users affected

End user, programmer, system analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The following library routines functionally comply with the XPG4 standard:

- abort(3)
- abs(3)
- acos(3)
- asctime(3)
- asin(3)
- assert(3)
- atan(3)
- atan2(3)
- atexit(3)
- atof(3)
- atoi(3)
- atol(3)
- bsearch(3)
- calloc(3)
- catclose(3)
- catgets(3)
- catopen(3)
- ceil(3)
- cfgetispeed(3)
- cfgetospeed(3)
- cfsetispeed(3)
- cfsetospeed(3)
- clearerr(3)
- clock(3)
- closedir(3)
- confstr(3)

- cos(3)
- cosh(3)
- crypt(3); DES encryption code requires a separate license for sites outside of the U. S. and Canada
- ctermid(3)
- cuserid(3)
- ctime(3)
- difftime(3)
- div(3)
- drand48(3)
- encrypt(3); DES encryption code requires a separate license for sites outside of the U. S. and Canada
- erand48(3)
- erf(3)
- exp(3)
- fabs(3)
- fclose(3)
- fdopen(3)
- feof(3)
- ferror(3)
- fflush(3)
- fgetc(3)
- fgetpos(3)
- fgets(3)
- fgetwc(3)
- fgetws(3)
- fileno(3)
- flockfile(3)
- floor(3)
- fmod(3)
- fnmatch(3)
- fopen(3)
- fprintf(3)
- fputc(3)
- fputs(3)
- fputwc(3)
- fputws(3)

- fread(3)
- free(3)
- freopen(3)
- frexp(3)
- fscanf(3)
- fseek(3)
- fsetpos(3)
- ftell(3)
- ftrylockfile(3)
- ftw(3)
- funlockfile(3)
- fwrite(3)
- gamma(3)
- getc(3)
- getchar(3)
- getcwd(3)
- getegid(3)
- getenv(3)
- getgrgid(3)
- getgrnam(3)
- getlogin(3)
- getopt(3)
- getpass(3)
- getpwnam(3)
- getpwuid(3)
- gets(3)
- getw(3)
- getwc(3)
- getwchar(3)
- glob(3)
- gmtime(3)
- hcreate(3)
- hdestroy(3)
- hsearch(3)
- hypot(3)
- iconv(3)

- iconv_close(3)
- iconv_open(3)
- isalnum(3)
- isalpha(3)
- isascii(3)
- isatty(3)
- iscntrl(3)
- isdigit(3)
- isgraph(3)
- islower(3)
- isnan(3)
- isprint(3)
- ispunct(3)
- isspace(3)
- isupper(3)
- iswalnum(3)
- iswalpha(3)
- iswcntrl(3)
- iswctype(3)
- iswdigit(3)
- iswgraph(3)
- iswlower(3)
- iswprint(3)
- iswpunct(3)
- iswspace(3)
- iswupper(3)
- iswxdigit(3)
- isxdigit(3)
- j0(3)
- j1(3)
- jn(3)
- jrand48(3)
- labs(3)
- lcong48(3)
- ldexp(3)
- ldiv(3)

- lfind(3)
- lgamma(3)
- localeconv(3)
- localtime(3)
- log(3)
- log10(3)
- longjmp(3)
- lrand48(3)
- lsearch(3)
- malloc(3)
- mblen(3)
- mbstowcs(3)
- mbtowc(3)
- memccpy(3)
- memchr(3)
- memcmp(3)
- memcpy(3)
- memmove(3)
- memset(3)
- mktime(3)
- modf(3)
- mrand48(3)
- nl_langinfo(3)
- nrand48(3)
- opendir(3)
- pclose(3)
- perror(3)
- popen(3)
- pow(3)
- printf(3)
- pthread_attr_destroy(3)
- pthread_attr_getdetachstate(3)
- pthread_attr_init(3)
- pthread_attr_setdetachstate(3)
- pthread_condattr_init(3)
- pthread_cond_broadcast(3)

- pthread_cond_destroy(3)
- pthread_cond_init(3)
- pthread_cond_signal(3)
- pthread_cond_timedwait(3)
- pthread_cond_wait(3)
- pthread_create(3)
- pthread_detach(3)
- pthread_equal(3)
- pthread_exit(3)
- pthread_getspecific(3)
- pthread_join(3)
- pthread_key_create(3)
- pthread_key_delete(3)
- pthread_mutexattr_destroy(3)
- pthread_mutexattr_getkind_np(3)
- pthread_mutexattr_init(3)
- pthread_mutexattr_setkind_np(3)
- pthread_mutex_destroy(3)
- pthread_mutex_init(3)
- pthread_mutex_lock(3)
- pthread_mutex_trylock(3)
- pthread_mutex_unlock(3)
- pthread_once(3)
- pthread_self(3)
- pthread_yield(3)
- putc(3)
- putchar(3)
- putenv(3)
- puts(3)
- putw(3)
- putwc(3)
- putwchar(3)
- qsort(3)
- raise(3)
- rand(3)
- readdir(3)

- realloc(3)
- regcomp(3)
- remove(3)
- rewind(3)
- rewinddir(3)
- scanf(3)
- seed48(3)
- seekdir(3)
- setbuf(3)
- setjmp(3)
- setkey(3); DES encryption code requires a separate license for sites outside of the U. S. and Canada
- setlocale(3)
- setvbuf(3)
- sigaddset(3)
- sigdelset(3)
- sigemptyset(3)
- sigfillset(3)
- sigismember(3)
- siglongjmp(3)
- signgam(3)
- sigsetjmp(3)
- sigwait(3)
- sin(3)
- sinh(3)
- sleep(3)
- sprintf(3)
- sqrt(3)
- srand(3)
- srand48(3)
- sscanf(3)
- strcat(3)
- strchr(3)
- strcmp(3)
- strcoll(3)
- strcpy(3)
- strcspn(3)

- strerror(3)
- strfmon(3)
- strftime(3)
- strlen(3)
- strncat(3)
- strncmp(3)
- strncpy(3)
- strpbrk(3)
- strptime(3)
- strrchr(3)
- strspn(3)
- strstr(3)
- strtod(3)
- strtok(3)
- strtol(3)
- strtoul(3)
- strxfrm(3)
- swab(3)
- system(3)
- tan(3)
- tanh(3)
- tcdrain(3)
- tcflow(3)
- tcflush(3)
- tcgetattr(3)
- tcsendbreak(3)
- tcsetattr(3)
- tdelete(3)
- telldir(3)
- tempnam(3)
- tfind(3)
- tmpfile(3)
- tmpnam(3)
- toascii(3)
- _tolower(3)
- tolower(3)

- _toupper(3)
- toupper(3)
- towlower(3)
- towupper(3)
- tsearch(3)
- ttyname(3)
- twalk(3)
- tzset(3)
- ungetc(3)
- ungetwc(3)
- vfprintf(3)
- wcscat(3)
- wcschr(3)
- wcscmp(3)
- wcscoll(3)
- wcscpy(3)
- wcscspn(3)
- wcsftime(3)
- wcslen(3)
- wcsncat(3)
- wcsncmp(3)
- wcsncpy(3)
- wcspbrk(3)
- wcsrchr(3)
- wcsspn(3)
- wcstod(3)
- wcstok(3)
- wcstol(3)
- wcstombs(3)
- wcstoul(3)
- wcswcs(3)
- wcsxfrm(3)
- wctomb(3)
- wctype(3)
- wcwidth(3)
- wordexp(3)

- y0(3)
- y1(3)
- yn(3)

2.10.8 *External variables comply with XPG4*

Users affected

End user, programmer, system analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The following external variables functionally comply with the XPG4 standard.

- errno
- optarg
- signgam
- stdin
- timezone
- tzname

2.11 POSIX threads (Pthreads) supported

Users affected

Programmer

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release provides support for an interface based on POSIX 1003.4a threads (Pthreads). This support does not include all the Pthread functions because of the late finalization of the standard. Future releases will support additional Pthreads functionality as appropriate.

The UNICOS 9.0 release provides the following functionality in compliance with Pthreads:

- Changes to the UNICOS multitasking process model to support a single-process multitasking group model
- Changes to UNICOS signal semantics to support the new process model
- Thread management functions
- Thread synchronization primitives

- Support for thread-specific data
- Reentrant functions

The changes to the UNICOS process model and its effect on signal semantics are discussed in detail in subsection 3.10, page 3–47. The changes described in that subsection are the result of moving the UNICOS system from a multiple-process to a single-process multitasking group model.

The remaining changes to the UNICOS operating system are the addition of new functions for the creation, manipulation, and support of user-directed threads of control. These functions provide capabilities similar in scope to existing UNICOS macrotasking functions.

The following are the primary new functions:

New function	Description
`pthread_create`(3)	Creates a new thread
`pthread_join`(3)	Waits for a thread to exit
`pthread_exit`(3)	Exits the calling thread
`pthread_mutex_lock`(3)	Locks a mutex
`pthread_mutex_trylock`(3)	Attempts to lock a mutex
`pthread_mutex_unlock`(3)	Releases a mutex
`pthread_cond_wait`(3)	Waits for a condition to be signaled
`pthread_cond_timedwait`(3)	Waits for a condition or time-out
`pthread_cond_signal`(3)	Signals a waiter for a condition
`pthread_cond_broadcast`(3)	Signals all waiters for a condition
`pthread_setspecific`(3)	Sets a thread-specific value
`pthread_getspecific`(3)	Gets a thread-specific value

The following new functions provide reentrant versions of existing functions:

New function	Nonreentrant equivalents
`asctime_r`(3)	`asctime`(3)
`ctime_r`(3)	`ctime`(3)
`getgrgid_r`(3)	`getgrgid`(3)
`getgrnam_r`(3)	`getgrnam`(3)
`getlogin_r`(3)	`getlogin`(3)
`getpwnam_r`(3)	`getpwnam`(3)
`getpwuid_r`(3)	`getpwuid`(3)
`gmtime_r`(3)	`gmtime`(3)
`localtime_r`(3)	`localtime`(3)
`rand_r`(3)	`rand`(3)
`readdir_r`(3)	`readdir`(3)
`ttyname_r`(3)	`ttyname`(3)
`strtok_r`(3)	`strtok`(3)

The following functions are provided to support atomic execution of multiple standard I/O routines:

New function	Description
`flockfile`(3)	Locks a file
`ftrylockfile`(3)	Attempts to lock a file
`funlockfile`(3)	Unlocks a file

Currently, the UNICOS system does not support the thread cancellation or `pthread_kill`() functions.

For more information, see the individual man pages for these routines.

2.12 Installation enhancements

The following subsections describe installation enhancements for the UNICOS 9.0 release.

2.12.1 *X Window System interface to the menu system supported*

Users affected

Administrator, operator, system analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

UNICOS 9.0 provides a point-and-click, X Window System-based interface to the UNICOS Installation/Configuration Menu System. Pull-down menus commonly used with windowed applications allow the user to see all the possibilities in one glance.

With the new interface, not only may the user see the current menu, but also a "Where am I" display, which gives the location of the menu within the hierarchy. Also, Help screens can now be shown at the same time as the actual menu and can also display the execution display of a particular action.

CRAY EL systems require an X Window System terminal to use this feature. The feature may be used to reconfigure a system, but may not be used during initial installation. CRAY J90 systems can also use the menu system's configuration window.

Related publications

- *UNICOS Installation/Configuration Menu System User's Guide*, publication SG-2412

2.12.2 *CD-ROM supported for UNICOS upgrades, revisions, and updates*

Users affected

Administrator, operator, system analyst

Supporting hardware

All Cray Research systems supported by this release that have a CD-ROM drive except CRAY J90 series

Initial release: UNICOS 8.3

CD-ROM is supported for upgrade, revision, and update releases. This feature is available only on systems having a CD-ROM drive.

In releases prior to UNICOS 9.0, CD-ROM is supported only for initial installations of base releases.

Related publications

- *UNICOS Installation Guide*, publication SG–2112

2.12.3 *Configuration conversion for restricted releases supported*

Users affected

Administrator, operator, system analyst

Supporting hardware

All Cray Research systems except CRAY EL series

Initial release: UNICOS 8.3

In releases prior to UNICOS 9.0, the UNICOS Installation/Configuration Menu System automatically converted a system configuration from a previous UNICOS base release to a new UNICOS base release (for example, UNICOS 8.0 to UNICOS 9.0).

Beginning in UNICOS 9.0, conversion capability is extended to restricted releases. The menu system allows conversion from a restricted release to a new base release (for example, UNICOS 8.3 to UNICOS 9.0) or from a base release to a restricted release (for example, UNICOS 9.0 to UNICOS 9.1). The CRAY J90 series will support this feature beginning with a UNICOS 9.0 update.

Related publications

- *UNICOS Installation Guide*, publication SG–2112

2.12.4 *Using* `ldcache` *during system builds*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The system builder can specify file systems for assignment to the logical device cache. This capability is applicable only for interactive (nonbatch) builds.

For more information, see the online help text associated with the following installation tool menu:

```
UNICOS 9.0 Installation/Configuration Menu System
   Build/Install System
```

2.13 Online diagnostics

The following subsections describe enhancements to online diagnostics for the UNICOS 9.0 release.

2.13.1 *Enhancements to* `ddms`

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems

Initial release: UNICOS 8.3

The UNICOS 9.0 release enhances the `ddms`(8) disk diagnostic and maintenance system with the following features:

- Addition of the `-a` option, which displays valid disk devices and their types as defined in the `/dev/ddd` directory.
- Addition of the `-E` option, which is used with the `read`, `write`, or `surf` action to specify the number of errors allowed before `ddms` aborts the action.
- Addition of the `aft` action, which allows for the creation or update of an `/etc/aft` file (ASCII flaw table file) for a specified device from the current unhideables table. The `-w` option for use with the `aft` action also has been added. The `-w` option writes the unhideable entries of the user flaw table for the specified device to the `/etc/aft` file.
- Addition of the `wrecc` action, which creates error correction code (ECC) errors on a specified device. The `wrecc` action reads the data and ECC on a specified sector, toggles the specified bits, and writes the data and ECC. The sector is then read to verify that an ECC error is generated.

- Enhancement of the `makeuft` action so that it creates a new User Flaw table on the target disk by reading and generating the flaw table information from the sector IDs instead of from the Factory Flaw table.

For more information on these enhancements, see the `ddms`(8) man page.

2.13.2 *New OLNET FDR-4 test*

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems configured with an IOS-E and FDR-4 hardware

Initial release: UNICOS 8.0.4/8.3

The OLNET online diagnostic network communications program can perform online testing of FDR-4 hardware. The test can detect and isolate problems in the fiber-optic communications link between a Cray Research mainframe and an SSD solid-state storage device.

For more information, see the `olnet`(8) man page.