GA0015

Fermilab

OCS Overview

Marilyn Schweitzer

06/27/95

Version 2.4

Abstract

Operator Communications Software (OCS) is a package that performs operator assisted tape mounts, tape drive allocation and tape drive tracking/management. This document gives an overview of OCS features.

Fermi National Accelerator Laboratory
Batavia, IL 60510

1.0 Introduction

Operator Communications Software (OCS) is a package that performs and manages tape drive allocations, operator assisted tape mounts and tape drive use statistics. It is most beneficial in a networked environment that has many shared taped drives and an operations staff responsible for mounting the tapes. OCS is not a tape I/O package.

It is written in C-language to run on UNIX operating systems and is currently supported on IRIX and AIX platforms. The underlying architecture uses SUN RPC for TCP/IP and UDP communication.

2.0 Interfaces

OCS provides interfaces from three different perspectives:

- Users may use OCS via the Fortran/C library, command line or X Motif tools.
- Operators use an X Motif tool to respond to tape mount requests.
- System Managers use a combination of Command line or X Motif tools to install and administer OCS. Additional command line or X Motif tools may be written by a System Programmer using any of the above building blocks.

The features from these perspectives are summarized in the following subsections.

2.1 User Perspective

2.1.1 Tape Drive Allocation

A user requests tape drive be allocated for his/her exclusive use. The allocation request may be either for a specific tape drive or for a tape drive from a logical group of drives and of a specific type. If no tape drives are available that meet the specified criteria, the user my request to queue until one is available. Examples of some allocation requests that OCS can handle are:

a) Allocate any tape drive on the local host.
b) Allocate any exabyte_8500 tape drive on the local host and if none are currently available, then queue until one is available.
c) Allocate any exabyte tape drive from the logical OCS group “mygroup”.
d) Allocate logical tape drive “mydrive”.
e) Allocate logical tape drive “yourdrive”, but if it is not available, then allocate any 9track drive from the logical OCS group “yourgroup”.

The allocation will not succeed if the user does not have OCS authorization to allocate the drive.

For further information, refer to the following subroutines and commands described in the "OCS Reference Guide":

Subroutines	**Commands**
ocs_alloc_cancel()	ocs_allocate
ocs_alloc_reply()	
ocs_allocate()	
ocs_no_preference()	
ocs_preference()	
ocs_queue_alloc()	

2.1.2 Mounting Tapes

Once a tape drive is allocated, the user sends a request to the operations staff specifying:

a) the logical tape drive name
b) the external tape identification (e.g. VSN)
c) whether the tape should be mounter read-only versus write-enable
d) optionally, a special message for the operator

After a request is sent, the user must fetch the mount reply by either:

a) polling to see if the request has been satisfied
b) waiting for a specific number of seconds for the reply
c) waiting forever for the reply
d) waiting for a signal is received

Multiple requests for different tape drives can be sent before retrieving a reply, but the replies may come in asynchronously. The user can then ask OCS to verify that the device is ready, has the correct read/write protection, and, optionally, verify the tape label.

For further information, refer to the following subroutines and commands described in the "OCS Reference Guide":

Subroutines	**Commands**
ocs_check_tape()	ocs_cancel
ocs_mount_cancel()	ocs_check_tape
ocs_mount_request()	ocs_request
ocs_mount_reply()	ocs_pending
ocs_use_sigio()	

2.1.3 Device Files

OCS maintains a mapping from the logical tape drive names to device files appropriate for reading or writing the particular device type. When the user is ready to open a device file for reading or writing, he/she can ask OCS for the device file with the appropriate attributes such as:

a) variable versus fixed length records
b) rewind on close versus no rewind on close
c) byte swapping versus no byte swapping
d) 8200 exabyte versus 8500 exabyte mode

The user can also set the ownership and protection on a tape drive, though this is usually done when the drive is allocated. On AIX tape drives, the user can also change the device characteristics.

For further information, refer to the following subroutines and commands described in the "OCS Reference Guide":

Subroutines	**Commands**
ocs_devfile_default()	ocs_devfile
ocs_devfile_get()	ocs_setdev
ocs_defvile_set()	ocs_setown
ocs_setdev()	ocs_logdev
ocs_setown()	
ocs_logdev()	

2.1.4 Dismounting Tapes and Deallocating Drives

When the user is done with a particular tape, he/she may dismount it and request another tape to be mounted. When the user is done with the logical drive, he/she should deallocate the tape drive for subsequent jobs and other users.

For further information, refer to the following subroutines and commands described in the "OCS Reference Guide":

Subroutines	**Commands**
ocs_deallocate()	ocs_deallocate
ocs_deallocate_all()	ocs_dismount
ocs_dismount()	ocs_left_allocated

2.1.5 Other User Features

Some other features a user may perform are:

a) Mark a tape drive broken so that it will not be allocated until it is repaired.
b) Log some statistics as to how much the drive was used.
c) Send an attention message to the operator.
d) Send a request to the operator to run a cleaning tape through a tape drive.
e) Display status of tape drives in the OCS database
f) Display pending mounts
g) Display tape drive statistics
h) Display tape mount log
i) Display tape drives that may need to be cleaned

For further information, refer to the following subroutines and commands described in the "OCS Reference Guide":

Subroutines	Commands	X Motif Tools
ocs_device_type()	ocs_broken	xocs
ocs_clean_it()	ocs_clean_list	xtapeview
ocs_init_stat()	ocs_clean_it	
ocs_mark_broken()	ocs_devstat	
ocs_oper_message()	ocs_init_stat	
ocs_report_stat()	ocs_message	
	ocs_report_stat	
	ocs_tape	

2.1.6 Examples

Packaged with OCS are example programs that demonstrate many of these features. There are examples written in FORTRAN, C-language and Bourne Shell. Users may use these examples as the basis for developing their own programs.

2.2 Operator Perspective

Operators may have multiple consoles serving tape drives from a single OCS database. An operator can respond to tape mounts with success, a pre-defined failure message or a customized failure message. When he/she responds to a mount on one console, the other consoles will be updated to show that the tape is no longer pending.

Operators may also be authorized to perform many of the System Manager functions described in the next subsection. For more information, refer to the "XTape" document and "OCS Reference Guide".

2.3 System Manager Perspective

2.3.1 Administration

The System Manager decides:

- which hosts can have access to tape drives through OCS
- which tape drives are to be controlled by OCS
- what logical name is assigned to each tape drive

This information is administered into a database maintained by OCS.

The System Manager then defines logically groups of tapes drives and which users are authorized to use tape drives from each logical group. For example:

a) Jack and Jill may be authorized to use any of the tape drives in the logical group “production”

b) Jack, Bob, and Ted may be authorized to use any of the tape drives in the group “test”

c) The logical group “test” may consist of the logical tape drives “tstdrv1” and “tstsdrv2” on host “cpuone”

d) The logical group “production may consist of all the logical tape drives on “cputwo” and only the logical taped drives “prod3” and “prod4” on host “cpuone”

There are two special groups:

“public”	Any user has access to the tape drives in this group.
“admin”	Any user added to this group has access to all the tape drives in the database and has permission for special functions such as adding and deleting tape drives in the OCS database.

For further information, refer to the following subroutines and commands described in the “OCS Reference Guide”:

Subroutines	**Commands**	**X Motif Tools**
ocs_list_tapes()	ocs_auth	ocs_install
	ocs_devstat	ocs_group
	ocs_terminst	xocs
	ocs_tape	

2.3.2 Reports

OCS keeps track of the following statistics for each tape drive:

- the serial # of the logical drive
- date drive was installed in OCS
- # of operator assisted tape mounts since its installation
- # of MB read and written since its installation
- # of errors since installed
- # of times the drive was cleaned since it was installed
- date the drive was last cleaned
- # of operator assisted tape mounts since the last cleaning
- # of MB read and written since the last cleaning
- # of errors since the last cleaning
- whether the tape drive's cleaning bit was set or not

The system manager may display this information via the following reports:

- operator assisted tape mount log
- currently pending tape mount requests
- tape drive use log
- tape drive cleaning history and drives which may require cleaning

In addition, the System Manager will receive a report via mail whenever a tape drive is marked broken, working, replaced or deleted. These messages may also be sent to specified users.

For further information, refer to the following subroutines and commands described in the "OCS Reference Guide":

Subroutines	**Commands**	**X Motif Tools**
ocs_list_reports()	ocs_clean_list	xocs
ocs_list_stats()	ocs_mrlog	
ocs_list_tapes()	ocs_reports	
	ocs_stats	
	ocs_tape	

2.3.3 Maintenance

OCS provides a variety of tools to display the status of a particular tape drive and tools to change the status, such as marking a tape drive broken. A tape drive which is marked broken will not be allocated to a user until it is marked working again or replaced.

OCS also provides tools for the system manager to back up the OCS database and ensure that the OCS log files are kept to a reasonable size.

For further information, refer to the following subroutines and commands described in the "OCS Reference Guide":

Subroutines	Commands	X Motif Tools
ocs_clean_it()	ocs_backup	xocs
ocs_mark_clean(	ocs_broken	
ocs_mark_broken()	ocs_cleaned	
ocs_mark_replaced()	ocs_clean_list	
ocs_mark_working()	ocs_clean_it	
ocs_list_tapes()	ocs_devfile	
	ocs_logdev	
	ocs_devstat	
	ocs_left_allocated	
	ocs_repaired	
	ocs_reports	
	ocs_reboot	
	ocs_tape	
	ocs_xfer	

3.0 Underlying Architecture

3.1 Database Server

OCS uses a relational database package developed at Fermilab. This database contains information on the tapedrives, hosts, mount requests, operator connections, mount history, tape drive statistics, etc.

Any number of hosts can share an OCS database, however, when the node with the database is down, none of the hosts that use the database can use OCS for tape mounts. The host with the OCS database needs to have a system daemon called **dbserv** installed. The **dbserv** uses TCP and is started by **inetd** when the first RPC call is issued to it. The **dbserv** is totally passive; it replies to RPC calls and does not initiate any RPC calls.

3.2 Tape Daemon

Another OCS system daemon which must be installed is called **taped** which also gets started by **inetd** when the first RPC call is issued to it. The **taped** uses TCP for except to communicate with a **taped** on an other host. In that case, UDP is used along with a retry algorithm internal to OCS.

The **taped** must be installed on each host which is going to issue OCS commands such as tape drive allocations, mount requests, mount replies, tape drive status, etc. The **taped** need not be installed on a host if the host is just going to be used for linking executables with OCS.

3.3 Device Daemon

The third and final OCS system daemon is called **devd**. The **devd** is started by the local **taped** on an as needed basis and uses TPC on a transient RPC number. The **devd** handles commands such as tape rewinds which can be time consuming and block until completed. At any given time there will be at most one **devd** per tape drive. If a **devd** has no activity for 10 minutes it will shut itself down.