

Road

To: Distribution
From: N. I. Morris
Date: February 4, 1975
Subject: I/O Interfacer Access to Disk

Overview

A means has been developed to allow the use of disk as an I/O device. This mechanism will enable a user ring program to access disk through the use of the I/O Interfacer in a manner similar to tape. When an appropriate user ring DIM is developed, user ring programs will be able to easily read and write a disk pack. In addition, on-line T&D will be able to attach a disk drive and perform tests. The mechanism will utilize either a separate disk subsystem or a subsystem shared with Multics page control. Allocation of disk drives on a separate, freestanding subsystem is straightforward and, in fact, identical to the mechanism used for tape. The sharing of a disk subsystem introduces complications which are discussed below.

Sharing a Disk Subsystem

A disk subsystem consists of one or two controllers connected to an IOM via one to four PSI links. Each link may consist of up to 8 logical channels. Thus, in effect, a disk subsystem is shared over these several logical channels. In most respects, these channels may be considered to be independent of each other. Connects may be issued to each of the channels with no software-visible interference between the channels.

In the disk access mechanism, one or more of the logical disk channels are reserved for use through the I/O Interfacer. These channels may be used to access disk drives not in use by Multics page control, with no software interference. The channels which are reserved for such use are carefully chosen to be lower in priority than those remaining in use by page control. Thus, user disk I/O should cause little or no degradation of disk paging throughput.

Disk I/O Initialization

User disk I/O initialization is performed at the end of Multics system initialization, long after disk paging has been enabled. At the time of disk I/O initialization, the channels needed are usurped from page control. Clearing a certain bit in the channel's control table causes page control to stop using that channel. The `iom_manager` is then called to unassign the channel, and it is reassigned through the I/O Interfacer. At that time, the disk drives to be used for disk I/O are checked against the drives in use by page control. If conflicts are found, system initialization is terminated.

Special Interrupts

DIMs which perform disk I/O through the I/O Interfacer need to receive the special interrupt which is generated when a disk pack is mounted and readied. The special interrupt, however, is sent only to the highest priority logical channel of a PSI link. This channel remains in use by page control. Thus, a mechanism is required to deliver special interrupts, which are unwanted by page control, to the I/O Interfacer. The I/O Interfacer is "made to think" that special interrupts are coming directly from the IOM channel.

New Configuration Card

A new configuration card is needed to define the channels and disk drives to be used for user disk I/O. The format of that card is shown below.

PRPH DISK iom chn model nchn sys max first n share

iom is the number of the IOM to which the disk subsystem is connected.

chn is the first IOM channel number which the disk subsystem uses.

model is the model number of the disk subsystem. Currently, the disk access mechanism is capable of supporting only the DSS191 disk subsystem. Hence, this parameter should be set to "191".

nchn is the number of logical channels to be used to access disk drives.

sys is the number of disk drives to be reserved for system functions (e.g. backup).

max is the maximum number of disk drives a user is allowed to have attached at any one time.

first is the drive number of the first disk drive to be used for user I/O disk access. Note that this number is the device number, not the area number.

n is the number of disk drives to be used.

share is an indicator of disk subsystem sharing with page control. If the disk subsystem used for disk I/O is freestanding, this parameter should be omitted. Otherwise, it should be set to the name of the disk subsystem being shared (i.e. "D191" or "E191").

Note that if disk subsystem sharing is indicated, the IOM channel numbers specified on the PRPH DISK card will be ignored. Channels will be assigned in order of lowest priority first as described above.

Calling Sequences

Entry: ioi_\$io_disk_attach

This entry is called to assign a disk drive and perform an attachment through the I/O Interfacer. A message is sent to the operator instructing him to mount the specified disk pack on the assigned drive.

Usage

```
declare ioi_$io_disk_attach entry (fixed bin, fixed
                                bin(6), char(*), bit(1) aligned, fixed
                                bin(71), fixed bin(35));
call ioi_$io_disk_attach (devx,      drive,    pack_id,
                          write_sw, ev_chan, rcode);
```

devx is the device index to be used in all subsequent calls to the I/O Interfacer. (Output)

drive is the assigned disk drive number. (Output)

pack_id is a character string identifying the disk pack to be mounted. (Input)

write_sw is a switch indicating whether or not the disk pack is to be written. If write_sw is "0"b, the operator will be instructed to

mount the pack write inhibited. (Input)

ev_chan is an ipc_ event channel ID to be used for wakeups when status is stored for the assigned disk drive. (Input)

rcode is a standard error code. (Output)

Entry: hphcs_\$io_disk_priv_attach

This entry is to be used by system functions such as backup. It will assign one of the disk drives reserved for use by system functions. If these drives are all in use, any free disk drive will be assigned.

Usage

```
declare hphcs_$io_disk_priv_attach entry (fixed bin,
    fixed bin(6), char(*), bit(1) aligned, fixed
    bin(71), fixed bin(35));
call hphcs_$io_disk_priv_attach (devx, drive, pack_id,
    write_sw, event_id, rcode);
```

Entry: ioi_\$io_disk_detach

This entry is called to detach an assigned disk and report the error count to the operator.

Usage

```
declare ioi_$io_disk_detach entry (fixed bin(6), fixed
    bin, fixed bin(35));
call ioi_$io_disk_detach (drive, errors, rcode);

errors is a count of the number of errors
encountered during disk usage. (Input)
```

Entry: hphcs_\$add_io_disk_drive

This entry provides the capability of removing and adding disk drives to the pool of drives available for attachment. Only drives removed through the use of this call can be added back at a later time.

Usage

```
declare hphcs_$add_io_disk_drive entry (fixed bin(6),
    bit(1) aligned, fixed bin(35));
call hphcs_$add_io_disk_drive (drive, sw, rcode);

sw is a switch indicating whether the drive is
to be added or removed. If it is "1"b, the
```

drive will be deleted. If it is "0"b, the
drive will be added. (Input)

(END)