To: Distribution

From: Richard Holmstedt (HVN 341-7248)

Date: 04/04/81

Subject: The new quota_monitor command.

INTRODUCTION

Running out of disk space (physical disk space or directory
quotas) is a problem which every Multics system must concern
itself with due to the disastrous impact it could have. This
bulletin outlines a new command called quota_monitor which would
provide a more automated means to safeguard against such
problems.

PROPOSAL

   As requirements for storage in the system_control_1 directory
are a result of system functions which for the most part are
self-acting, it has been hard to gauge storage needs. When quota
in this directory becomes exhausted it usually results in a
system crash due to "attempt to terminate Initializer process".
For this reason a prototype quota_monitor command was written to
provide a warning of approaching RQO condition.   Its use on
system M has proven effective in preventing crashes as a result
of this problem.

   The use of quota_monitor would not be limited only to the
system_control_1 directory.   The command is intended to provide
monitoring of any terminal quota directory where exhaustion of
storage would be a problem.   The command can be invoked in any
process with sufficient access, but is intended to be run in a
SysDaemon process which would provide continous service.

The quota_monitor command when invoked will call
timer_manager_$alarm_call with a time interval dependent on how
much available storage was found. That is, if the directory was
50% full, then an alarm would be set to trigger in 30 minutes to
check again. If the quota was found to be at 80% then a message
would be sent to the system console and an alarm time of 5
minutes would be set. At 90% it would send a warning to the
system console every minute, and increase the quota by an
additional number of records, if storage on the logical volume is
available.

When quota for a directory is checked, the storage remaining for
use on the directory's logical volume will also be calculated.
If the logical volume is found to be running low on usable
storage, then a warning would be sent to the systems console.
The task of monitoring storage is usually the responsibility of
the System Administrator. To monitor quota the Administrator
would only need to determine which directories are to be
monitored and allow the system to provide this function.

Some suggestions for implementation have been to let the
Initializer handle the RQO. To have an RQO handler in the
Initializer would be unreasonable due to the added overhead and
because it could be the Initializer who was causing the problem.
The use of page control was also considered, but to put the
monitor within page control would create unacceptable overhead
and limit a site's control of its use. For these reasons, a
separate SysDaemon process should be used to watch for unsafe
exhaustion of storage. This would be an important service that
would improve system reliability and management of system
storage.

    Comments to this proposal should be sent by Mail Facility to
Holmstedt.SysMaint @ system M.

quota monitor

Syntax as a command:

        quota_monitor {-control_arguments}

Function: Calculates storage of a terminal quota directory and will send a warning message to the system console at the approach of a record quota overflow condition. The storage remaining on the directories logical volume will be calculated and a warning issued if critically low on space.

Argument:

Control arguments:

-path, -pn

Pathname of the directory to be monitored.

-add_quota N

This argument will indicate the number (N) of pages to be added when a directory quota becomes critically low. If storage is not available on the logical volume then none will be added and a warning message will be issued. If omitted, then only a warning of the approaching RQO will be issued.

Note: Access to this command requires phcs_ access to issue warnings on the system console. Access to hphcs_ is required if the -add_quota argument is used.