GIMIX RAMdisk

Virtual Disk Driver
for OS-9 GMX II and GMX III

GIMIX Inc.
1337 W. 37th Place
Chicago, IL 60609
312-927-5510 * TWX 910-221-4055

GIMIX RAMdisk V1.05
for OS-9™

INTRODUCTION

GIMIX RAMdisk is a set of programs that convert a portion of the Random Access Memory (RAM) in an OS-9 GMX II or OS-9 GMX III system into a high-speed mass-storage device or RAMdisk.

This "virtual disk" is used in exactly the same manner as a "physical device" such as a floppy or hard disk. Files are created or accessed by including the device name of the RAMdisk in an OS9 pathlist. Throughout this manual, examples assume the default device name "R"; however, the name may be user-defined when the device descriptor is created.

Since it does not suffer from any mechanical restrictions on access time, the RAMdisk is considerably faster than a conventional disk. The capacity of the RAMdisk is user-defined and limited only by the amount of memory in the system.

Two OS-9 modules are required to create a RAMdisk. The driver module, a Random Block File (RBF) type device driver that handles data transfer between memory and the operating system, and a device descriptor module that defines the parameters of the "disk". Device descriptors are created by the user, using a simple utility program.

Three programs are included on the distribution diskette. The device driver module "Rdisk", and the utility programs "Rmake" and "Rrm". Rmake generates device descriptors for the RAMdisk, and Rrm is used to "remove" the RAMdisk in order to recover the memory allocated to it.

CONTENTS

SECTION 1: Installation

Copy the files Rdisk, Rmake, and Rrm to the normal execution directory (usually /d0/cmds or /h0/cmds). Note: The distribution diskette may include separate versions of one or more of these files tailored to a specific CPU board. See the "ReadMe" file on the diskette to determine the correct version for the CPU in the system.

There are two ways that the RAMdisk can be installed in the system. The first, and easiest, is to use the Rmake utility to build one or more files containing both the driver module and a device descriptor (see the Rmake documentation). Several such files can be created, each configured for a different size RAMdisk. One of the files is then be loaded into memory using the OS-9 "LOAD" utility. After loading an appropriate file, use the OS-9 "DIR" command (dir /r) to list a directory of the "disk". This step verifies that the modules have loaded correctly, and attaches the RAMdisk to the system for future use. The RAMdisk is accessed in the same manner as any other RBF type device, by including the device name in an OS-9 pathlist.

Note: When using this method of installing the RAMdisk, the driver and device descriptor should be loaded from a single file containing both modules. If the modules are loaded from separate files, the RAMdisk can be accessed; however, the CHD/CHX commands will not work properly with it.

***** IMPORTANT *****

Before writing any data on the RAMdisk, you MUST do a "DIR" of the device or change the working data directory to it (chd /r), to "format" the RAMdisk and attach it to the operating system. A separate "format" program is not required by the RAMdisk. Once initiallized in this way, RAMdisk will remain attached until the system is reset or the Rrm utility is used to remove it.

****

The second way to install the RAMdisk, is to include the device driver module (Rdisk) and a device descriptor (created by Rmake) in the OS9Boot file on the system boot disk. After the system is booted, a "dir" or "chd" will attach the RAMdisk as described above.

The RAMdisk device driver is only capable of supporting one "drive" at a time. Only one device descriptor and RAMdisk can be in use at any one time. To alter the size of the RAMdisk you must "remove" the existing disk and install a new one, with a descriptor configured for the required size.

SECTION 2: Disk Size and Memory Requirements

The size of the RAMdisk (and the amount of memory it requires) is user-defined by the number of sectors specified when creating a device descriptor with Rmake. Each sector is 256 bytes long and the RAMdisk requires [256 x sectors] bytes of RAM in addition to the RAM required for the driver and descriptor. The system must have sufficient memory for the RAMdisk, plus enough free memory for programs that will be run while it is in use. Some trial-and-error testing may be necessary to determine the largest RAMdisk that can be used, while retaining enough free memory for other requirements.

OS-9 allocates memory to the RAMdisk in blocks, of a size determined by the Memory Management hardware on the CPU board. The number of sectors on the RAMdisk should be an integral multiple of the number of sectors that will fit in one block (8 for OS-9 GMX III (GMX III CPU) or 16 for OS-9 GMX II (CPU Plus)). If the number of sectors on the disk is not an integral multiple of the sectors per block, part of one block will be unused and wasted.

SECTION 3: Removing the RAMdisk

The utility program Rrm is used to "remove" the RAMdisk and return its memory to the system. If the RAMdisk driver and descriptor were loaded from disk, they are unlinked and their memory is also returned. If they were loaded as part of the boot file they are unlinked, but their memory is not returned. In either case, all of the memory allocated to the disk itself is returned to the system's free memory pool.

***** CAUTION *****

Rrm UNCONDITIONALLY removes the RAMdisk, even if it is in use. In addition, when the RAMdisk is removed, any files it contains are destroyed and can not be recovered. Before using Rrm to remove the RAMdisk, all open paths to the disk must be closed, and the files must be copied to a conventional disk if is necessary to preserve them.

Use extreme caution when removing a RAMdisk, especially on a multi-user systems. Be sure there are no open paths to the device and that the files have been copied if necessary.

Access to the Rrm utility should be restricted to prevent unauthorized or unintentional use.

SECTION 4: Installation Example

The following example illustrates the step-by-step installation of RAMdisk in a typical system. It assumes that the programs "Rdisk", "Rmake", and "Rrm" have been copied to the current execution directory (/h0/cmds in this example). The RAMdisk created will contain 512 sectors and require 128K of free RAM in addition to the RAM required by the driver/descriptor.

```
                                                    Comments

OS9: copy /h0/cmds/Rdisk /h0/cmds/R128K          * make a copy of the driver

OS9: Rmake R128K 512                             * make 512 sector descriptor
                                                   and append it to the copied
                                                   driver

OS9: load R128K                                  * load the driver/descriptor
                                                   into memory

OS9: dir /r                                      * directory of RAMdisk to
                                                   initialize and attach it
  directory of /r    09:45:51

OS9: free /r                                     * display RAMdisk free space

  "GIMIX RAM Disk" created on: 84/06/06
  Capacity: 512 sectors (1-sector clusters)
  505 free sectors, largest block 505 sectors

OS9: copy anyfile  /r/anyfile                    * copy a file to the RAMdisk

OS9: chd /r                                      * change data directory to
                                                   the RAMdisk

OS9: dir                                         * see if it copied ok

  directory of .     09:46:44

  anyfile

OS9: Rrm                                         * remove the RAMdisk and
                                                   release its memory
   Purge RAMdisk device /r (y/n) ? y               Note: This destroys any
                                                   files on the "disk"
OS9:
```

RMAKE

Create RAMdisk device descriptor.

SYNTAX: RMAKE <file> <sectors> [<module name>] [<module revision>]

FUNCTION: RMAKE is used to generate device descriptors for the GIMIX RAMdisk, virtual disk driver (RDISK). It allows the user to tailor the size of the RAMdisk to the application and the available memory.

The first parameter <file> is the filename for a file that will be created by RMAKE in the current execution directory. If a file by this name already exists, the descriptor will be appended to the file.

<sectors> is the number (in decimal) of 256 Byte "sectors" that will be available on the RAMdisk. The minimum number of sectors is 8, the maximum is limited by the amount of available RAM. For optimum memory utilization, the number of sectors requested should be a multiple of 8 for OS-9 GMX III (CPU III) or 16 for OS-9 GMX II (CPU Plus).

The remaining two parameters are optional; however, if the second is used, the first must also be given. The first option [<module name>] is a user-selected module name for the descriptor and is also the device name that will be used in OS-9 pathlists to access the RAMdisk. If none is given, the default name is "R". The second option [<module revision>] can be any number from 1 to 15 (the default is 1), which becomes the module revision number in the Attributes/Revision byte of the descriptor.

EXAMPLES:

```
OS9: RMAKE Rdesc 512


OS9: copy /h0/cmds/rdisk /h0/cmds/r400
     RMAKE r400 400
```

In the first example, assuming that the file "Rdesc" does not already exist in the current execution directory, RMAKE creates a file "Rdesc" which contains a device descriptor module configured for 512 sectors, with the default module name "R" and the default revision number (1). The descriptor, along with the device driver (Rdisk) can then be included in a bootfile (OS9boot) using the OS9GEN utility.

In the second example, a copy of the device driver module (rdisk) having the name r400 is made in the execution directory. RMAKE is then used to create a 400 sector device descriptor and append it to the copy of the driver. The resulting file (r400) contains both the driver and descriptor.

NOTE: If the device driver and descriptor are not loaded as a unit, either by including both in the bootfile or by loading a single file containing both modules, the change directory commands (CHD, CHX) will not work for the RAMdisk.

See the RAMdisk manual for more information.

RRM

Remove RAMdisk and release its memory

SYNTAX: RRM [<device name>]

FUNCTION: RRM is used to remove an established RAMdisk from the system and return its memory to the free memory pool.

The optional parameter <device name> is the module name of the device descriptor for the RAMdisk to be removed. If no name is given, RRM uses the default name "R".

CAUTION: RRM unconditionally removes the RAMdisk, destroying any files it contains. RRM must not be used unless it is certain that the RAMdisk is not in use and all open paths to the device have been closed. Files on the RAMdisk should be copied to floppy or hard disk if necessary, before the RAMdisk is removed.

EXAMPLES:

```
OS9: RRM

     Purge RAMdisk device /R (y/n) ?

OS9: RRM  /RD

     Purge RAMdisk device /RD (y/n) ?
```

In the first example, no device name is given. If the response to the prompt is "y" or "Y", RRM will remove the RAMdisk default device "R". Any other response causes RRM to abort without further action being taken.

In the second example, RRM will remove the device called "RD". Note: The leading "/" is optional when specifying the device name. RRM will add the "/" if it is omitted in the command line.