

Mark5A Software

Introduction

These are progress notes on the Mark-5A software in C for Linux. The Mark-5A commands and queries are described in Alan Whitney's (ARW's) memo "Preliminary Mark5A command set," cited in detail below. The program `Mark5A` parses these commands and calls the appropriate Conduant XLR `StreamStor` functions (cites below), to carry out the various operations, and returns replies as prescribed in ARW's memo. At this writing (date below), we have a beta version of these XLR API commands for Linux and revised versions of the `StreamStor` board, with which we have done initial development. The initial version of the software, `Mark5A` and some associated programs, with an initial version of the hardware system, designated `Mark5P`, has been released to selected users. We are continuing development of both hardware and software.

Implementation

Inside `Mark5A`, the function `Parse5A()`, contained in `Parse5A.c`, as the name implies, parses the commands and queries, `Parse5A()` then calls appropriate XLR functions. `Parse5A.c` also contains empty stubs for two functions, `outBoard()` and `inBoard()`, to talk with the output and input boards. This functionality, now combined into a single board, is only partly designed, and the software interface is not yet completely defined, so, at this writing, these functions can only be stubs.

`Mark5A` in `Mark5A.c` is a main program to call `Parse5A()`. This operational program accepts Mark-5A commands and queries from either RS-232 serial (`/dev/ttyS0` alias `COM1`) or sockets spawned from `m5drive tcp` (cf., `/etc/services`). The replies from `Parse5A()` are returned to the stream from which the commands or queries came. Some optional debuggery prints to the controlling terminal.

A test program, `tstMark5A` in `tstMark5A.c`, can be run on the same computer as `Mark5A` or on a different computer such as `ccc`. It connects to

an m5drive tcp socket on a prescribed computer with Mark5A running, accepts Mark5A commands and queries typed into the controlling terminal, sends these through the socket to Mark5A, and prints replies from Mark5A as read from the socket. The socket configuration in `tstMark5A.c` can be used as a model for other programs that connect to Mark5A's socket. The Haystack correlator also has a suite of functions to perform such tasks.

The file `stubs5A.c` contains stubs in lieu of the Conduant-supplied XLR functions called by these programs. This allows some preliminary testing at, for example, sites that don't yet have Mark-5 hardware.

Installation

Following are notes on how to install the Mark-5A programs on a Linux box. These instructions were tried on Red Hat Linux version 7.1. We assume, of course, that you already know all about Linux.

To begin, you'll need a login and directory for yourself, and you'll need to know the root password because some of these operations must be performed by root.

(1) Create subdirectories `C` and `bin` in your home directory. Set your path to include this `bin`.

(2) Copy all the Mark-5A files from a computer with Mark5A already installed into your directory. For example:

```
cd C
rcp -p " mark5-05:/home/jball/C/*5A* " .
cd bin
rcp -p mark5-05:/home/jball/bin/cc5A .
rcp -p mark5-05:/home/ball/bin/End5A .
```

This `cc5A` file is in lieu of a proper make file; `End5A` is to shut down Mark5A. The latest versions of all these files are also available from <http://fourier.haystack.edu/Mark5/>.

(3) Install the StreamStor (SS) software. The instructions for this were taken from an email from Cindy Gold at Conduant Corp. This assumes that you have access to the latest version of the StreamStore files. These can also be copied from a computer with Mark5A already installed. In mark5-05, for example, they are in /home/streamstor. And copies of the original SS tar files are now also in <http://fourier.haystack.edu/Mark5/>.

(A) Create a directory to hold StreamStor files and subdirectories. We use /home/streamstor:

```
mkdir /home/streamstor
cd /home/streamstor
```

(B) Copy or move the StreamStor files into this directory. The original files from Conduant have names that change with version. They usually come as tarred and gzipped files with names such as ss52.tgz.

(C) These need to be gunzipped and untarred to produce a directory with lots of stuff therein. Sdk will be a symbolic link to this directory changed, if necessary, with each new version. An example of all this is in mark5-05 as noted above. In the /home/streamstor directory, extract the StreamStor files. For example:

```
tar xvzf ss52.tgz
```

This should produce a subdirectory with a name such as linux with lots of stuff therein. Create a symbolic link, Sdk, to it, for example:

```
ln -s Sdk linux
```

(D) Cindy warns to check at this point that the Linux kernel source is installed on this machine and that the kernel-source-include directory is /usr/src/linux/include. If not, edit makefile in Sdk/driver. We had to change INCLUDEDIR in Sdk/driver/makefile to /usr/src/linux-2.4/include, but this is probably peculiar to our particular Linux installation. Also check Sdk/driver/setup_inst_dir and Sdk/driver/wdreg, which need execute permission; chmod 0755 ... if necessary.

(E) In Sdk/driver directory, run:

```
make install
```

You'll need to be root. If you see a problem with missing kernel entry points, try:

```
make clean
make
make install
```

Redoing the make this way seems to be necessary whenever the kernel is not just the same as Cindy used.

(F) Follow the instructions that print at the end of the operation of this make install. Perhaps easiest is to copy the last few lines from `/etc/rc.d/rc.local` on a working Mark5A machine into the same file on the new machine. This is needed on reboot.

(G) Set environment variable `STREAMSTOR_BIB_PATH` to point to `/home/streamstor/Sdk/bib` or wherever the bib files are. You should put this into your `.login`, `.cshrc`, or `.bashrc` file so that it happens each time you log in. Example:

```
setenv STREAMSTOR_BIB_PATH \  
/home/streamstor/Sdk/bib
```

Warning: Different versions have different bib paths.

(4) These are some additional details to allow the Mark-5A programs to compile and run and to allow you to use the StreamStor utility programs.

(A) Put symbolic links in your C directory as follows:

```
cd C  
ln -s /home/streamstor/Sdk/lib/libssapi.a .  
ln -s /home/streamstor/Sdk/include/xlrapi.h .  
ln -s /home/streamstor/Sdk/include/xlrtypes.h .
```

(B) Compile the Mark-5A programs using `cc5A`. Watch for errors and warnings. Check that you now have `Mark5A`, `tstMark5A`, and `Mark5A.stubs` in your `bin` directory.

(C) Put symbolic links in your `bin` directory to `ssopen` and `sstest`, which should be in `/home/streamstor/Sdk/util`. For example:

```
cd bin  
ln -s /home/streamstor/Sdk/util/ssopen .  
ln -s /home/streamstor/Sdk/util/sstest .
```

(D) If you plan to use your RS-232 serial port, set permissions (you'll need to be root):

```
chmod 0666 /dev/ttyS0
```

(E) Edit `m5drive 2620/tcp` and `m5data 2630/tcp` into `/etc/services`. These can be set to any numbers not otherwise in use, but they need to be the same numbers on any other machine that will talk with Mark5A using sockets. An easy way is to copy the last few lines from

/etc/services on a working Mark-5A machine. Note that the number recommended for `m5drive` by ARW in the VSI-S Draft 6.0 is 5653 rather than 2620.

Operation

With all the StreamStor hardware (card and disks) installed, you probably should run `ssopen` first. If it's not happy, then `Mark5A` is not likely to be happy. If `ssopen` works, then try `sstest` also, but `sstest` will try to *write over any SS data on the disks* and leave some 33 Mbytes of test data that you may want to erase.

There are a number of options available when using the Mark-5A programs. The test program, `tstMark5A`, for example, can run on the same machine as `Mark5A`, on a different machine, or, if you have another program that speaks Mark-5A, such as the Field System (FS), then maybe not at all. And `Mark5A` can run with various quantities of debug prints. If you don't have a StreamStor card and disks, then you might want to run `Mark5A.stubs` instead, but, of course, it can't really read or write any data; it just pretends.

Try:

```
Mark5A n &
```

where *n* is *msglev* from the following list:

msglev = -1 A vast quantity of debuggery prints

msglev = 0 Some debuggery prints

msglev = 1 Default and normal operation; warnings and errors print

msglev = 2 Only errors and operational messages print

msglev = 3 Nothing prints except fatal errors when the program dies

Omit *n* to get *msglev* = 1. You may also omit the `&` provided that you want to dedicate this terminal or window to `Mark5A` and run any other programs, including `End5A` shutdown, in another window. Then:

```
tstMark5A machine
```

will connect to `Mark5A` on *machine* using a socket and will accept Mark-5A commands and queries as described in "Preliminary Mark5A command set" cited below. Omit *machine* to take the default machine name: `localhost`.

This test program, `tstMark5A`, should be able to run on any Linux or Unix machine for which it has been compiled.

`Mark5A` can also be compiled with an option that will allow it also to accept commands on RS-232 serial port `ttyS0`. The baud rate defaults to 4800, no parity.

Shut Down and Change Disks

The StreamStor system is persnickety especially with respect to shutting down properly and swapping disks. The proper `Mark5A` shut-down procedure is as follows:

(1) First close all socket connections by ending any programs that have socket connections to `Mark5A`. For `tstMark5A`, a `<Ctrl>C` (interrupt) works; for the FS, try `mk5close`, which might already be in your FS schedule. For the correlator, you'll need to `endall` (sorry). If you fail to do this *first*, then you might find that socket `m5drive` is in limbo and can't be used again until after a reboot.

(2) Then
`EndM5`

This script sends an interrupt to the main thread of `Mark5A` only. Do *not* use just `<Ctrl>C` to `Mark5A` because this sends interrupts to all threads and creates a mess. (Just `<Ctrl>C` works for `Mark5A.stubs`.)

The commands `reset=dismount` and `reset=mount` are intended to allow hot swapping disks without shutting down `Mark5A`. Although not completely reliable, this usually works as described in ARW's memo. If this fails, then try shutting down `Mark5A`, `SSReset` (described below), then begin again with `ssopen` and `Mark5A`. If all else fails, try a power-down reboot.

The rules for installing disks, copied from the StreamStor manual, are: "If any drive cable has a slave drive, then all [in-use] cables must have a slave drive. Connectors on the controller board should be used in order without gaps starting from connector zero (0)." "Drive cable," and "connector" mean bus. Each slave must have a master. If any master has a slave, then all masters must have slaves. `Mark5A` checks that these rules are followed.

Auxiliary Programs

In addition to `ssopen`, `sstest` (from Conduant), and `tstMark5A`, mentioned above, there are several other possibly useful auxiliary programs to accompany the Mark-5 system. Following is a brief description of these.

`tstMark5A [machine]`

where *machine* defaults to `localhost`. `Mark5A` must already be running on *machine*, but `tstMark5A` can run on some other machine. This program accepts commands and queries as defined in ARW's memo, sends them to `Mark5A` on *machine*, and prints the responses.

`SSErase`

`Mark5A` must *not* be running. This program erases all data on SS disks. This is useful to test the write-ability of disks and to prepare disks to be sent to stations to be recorded.

`SSReset`

`Mark5A` must *not* be running. This program performs an `XLRCARDReset ()`, which often helps extricate the system from a no-fair state.

`DirList`

reads the Mark-5 directory and lists the contents including the starting and ending byte numbers of all the completed scans. `DirList` can run simultaneously with `Mark5A`, in which case the listing will be up to date except for any scan being recorded at the time. If `Mark5A` is in dismount mode (`reset=dismount`), then the listing will be appropriate for the SS disk(s) that were in place before the dismount. Or if `Mark5A` is not running, then the listing will be appropriate for whatever was happening last time `Mark5A` was running. This is equivalent to saying that `DirList` does not read SS disks, instead it reads whatever `Mark5A` has written into the directory.

Following are notes about two stand-alone programs intended to exchange data with Mark-5 machines and primarily intended to run on computers other than Mark-5 machines. The source code contains hints for compiling on Linux and HP-UX.

```
Net2file [ filename ]
```

accepts a connection from a Mark-5 machine and writes the received data to *filename* or to *save.data* if *filename* is blank. This file will be created if necessary or appended. Start `Net2file` *first*, then command `disc2net` or `in2net` in the Mark-5 machine. Monitor progress with

```
ls -l filename
```

but this will lag the actual progress because of buffering. `Net2file` will end when the Mark-5 machine disconnects or with `<Ctrl>C`, after which *filename* will be ready to use.

```
File2net machine [ filename [ startbyte [ endbyte ] ] ]
```

sends a file or part of a file to a Mark-5 machine. Command `net2disc` or `net2out` in the Mark-5 machine *first*, then start `File2net`. *Filename* defaults to *save.data*, *startbyte* defaults to 0, and *endbyte* defaults to the end of the file. `File2net` ends when the prescribed transfer is done.

References

“Preliminary Mark5A command set (Revision 1.8),” by Alan R. Whitney, 19 September 2002. (<ftp://web.haystack.edu/pub/mark5/008.8.pdf>).

“Mark 5A Input/Output Board Control/Status,” by Alan R. Whitney, 4 October 2001. (<ftp://web.haystack.edu/pub/mark5/009.pdf>).

“Installation and User’s Guide, Conduant StreamStor Real-Time Storage Controller,” Conduant Corporation. (http://www.boulderinstruments.com/technical_pdf/ssmanual.pdf, 4/26/02, or <http://fourier.haystack.edu/Mark5/StreamStorManual.pdf>, 9/30/02).

“StreamStor Software Enhancements Plan (preliminary),” Conduant, 1/10/2002. ([API_5.X_Additions.doc](#)).

Revised: 2002 November 3, JAB