JET PROPULSION LABORATORY

To: Time Warp Folks

Re: Some Man Pages

From: Steven Bellenot

The following Man pages are (very) brief "how to use" documentation for some of the software either modified or related to code modified by myself. The page on m3plot is not complete, but the menu item "walkin" this described. Like all software, this is given "as is" which means "I think it is OK, but something always gets overlooked, changed or otherwise abused". Enjoy.

Man pages:
exstat, hc, hgram, history, m3plot, msgstat, tics.

exstat [-pN] [filename]

Exstat does for execution intervals what msgstat does for messages. Exstat reads from the files cname and cflow and outputs to ex.his. If the optional filename "foo" is included on the command line, exstat reads from the files foocname and foocflow. The output file ex.his is a histogram of all execution times. All execution intervals over "Publish Limit" are displayed to stderr.

OPTIONS:
    -pN   where N is a double, sets the Publish Limit in
    milliseconds, the default is 200.0.
SEE ALSO:
    history, msgstat.
FILES:
    exstat.c, history.c

hc doc [filename]

Hc (hyper-circle) is a graphs program for the Iris. Hc displays the hypercube of dimension "doc" and the I/O channels connecting the nodes. Hc reads its input from mname and mmlog (foomname and foommlog if the filename "foo" is on the command line). Hc shows the message paths of all messages in transit at the same time. The right button advances one message at a time, the middle button is fast forward mode and the left button exits the program.

Mex must be running for hc to operate correctly. Hc also needs the file tics to get correct times.

SEE ALSO:
        m3plot, tics
FILES:
        hc.c, indata.c

Man Page

hgram [-pN -uN -lN] [filename [.. filename]]

Hgram (histogram or frequency table) reads doubles written in ascii from standard input or a number of text files and outputs a histogram. Hgram is a (constant time) associative array over a reasonable range of values. Hgram will rescale its "bucket size" on the fly if needed. Hgram was designed for exponentially distributed data and is not suitable for general data. Hgram will reject negative values and it may run slow if there is lots of data at the high end.

Hgram will output a table in a form suitable for Excel and with tens of thousands of input values, hgram will often produce somewhere near 100 lines of output. (Excel's arbitrary limit on the number of data points it will plot.)

EXAMPLES:
If file data1 contains: 1.9 5 0.3 2.1 4.5
and file data2 contains: 1 1.4 0.9,
then "hgram data1" will produce:

0	1
1	0
2	2
3	0
4	0
5	2

and "hgram data1 data2" will produce:

0	1	0
1	0	3
2	2	0
3	0	0
4	0	0
5	2	0

OPTIONS:
-pN   where N is a double, sets the precision of the histogram, the default is -p1.0. No space between p and N.
-uN   where N is a double, sets the upper percentile of hgram's output, the default is -u100.0. No space between u and N.
-lN   (lower case "L") where N is a double, sets the lower percentile of hgram's output, the default is -l0.0. No space between l and N.

MORE EXAMPLES:
"hgram -p0.5 data1" will produce:

0.0	0
0.5	1
1.0	0
1.5	0
2.0	2
2.5	0
3.0	0
3.5	0
4.0	0
4.5	1
5.0	1

while "hgram -p2.0 data1" will produce:

0	1
2	2
4	1
6	1

Also "hgram -l25.0 -u55.0 data1" will produce

2	2

The message "history contracting" is given when the given precision is too small. The program quickly doubles the precision parameter and goes on. Roughly speaking there need to be over 500 values over 64 * precision for hgram to contract.

SEE ALSO:
history, msgstat, and the Excel macro file Hist.MS

```
history
h_percision ( double precision, history *target)
h_insert ( double value, history *target)
dump_history (history *target, FILE *outfile)
include file "history.h"
```

History is an implementation of an associative array of positive reals in c. Each "history variable" x needs to be initialized with a precision p. Values from 0.0 to $64*p$ can be inserted with constant time, values above $64*p$ are inserted into a sorted array. If there more than 512 values above $64*p$ the history will contract. History contracting is equivalent initializing with $2*p$ and inserting the same values. The non-destructive routine dump_history will output the histogram of the "history" target to file pointed to by outfile. (64 and 512 are of course #defines.)

History has other features and functions. In particular an array of historys can be used to collect related information. Also an attempt has been made to provide "safe" functions that look into the history structure. It is "almost" an ADT. Note there is no allocate_history function.

SEE ALSO:
        hgram, msgstat.
FILES:
        history.c, history.h.

m3plot [file1] [file2] [step] [file3]

M3plot is a graphs program for the Iris. M3plot displays all the messages in graphical format of real time vs virtual time. M3plot reads its input from mname and mmlog (foomname and foommlog if "foo" is file1 on the command line). If additional filenames are on the command line, m3plot will plot them all on the same graph. The messages from file2 will be shifted up "step" virtual time units. The default for step is 10, and step must be present to get the file3 inputted as well.

M3plot is a pop-up menu driven program. However the menu item "walkin" uses the following keys. The motion of the mouse if the X-direction while certain mouse buttons are down will move the graphic on the screen. In T mode (the default) the right button controls the in-out direction, the middle button the up-down direction and the left button the left-right direction. The "A" key will put you in angle mode, rotating left-right or up-down depending on the button which is down. The "R" key resets, the "T" key puts you back in translate mode, and the "Q" key quits you from walkin. (The F and D keys also do things.)

Mex must be running for m3plot to operate correctly. The file tics is needed to get the times correct.

SEE ALSO:
    hc, tics

msgstat [-cghmq] [filename]

Msgstat inputs the files [filename]mname [filename]mmlog produced by mstat and the file "tics" and produces a large number of files. Note that filename, if persent is glued infront of both mmlog and mname. Msgstat is best run in a near empty directory. Msgstat uses the histogram "associative array" of history.c to condense message data from tens of thousands to around a hundred data points.

FILES OUTPUTED:

chan.use: The number of messages using each channel of the hypercube. (Suppressed by the 'c' option.)

twtw.all: Histogram of Time Warp to Time Warp time. (Suppressed by the 'h' option.)

twhg.all: Histogram of Time Warp to Mercury time. (Suppressed by the 'h' option.)

hgtw.all: Histogram of Mercury to Time Warp time. (Suppressed by the 'h' option.)

hghg.hops: Histogram of Mercury to Mercury times by the number of hops. (Suppressed by the 'h' option.)

packs.hops:Histogram of Mercury to Mercury times by the number of Mercury packets. (Suppressed by the 'h' option.)

count.dlen:Histogram of messages processed before a waiting message in Mercury's receive queue. (Suppressed by the 'q' option.)

count.queh:Histogram of messages queued before a waiting message in Mercury's receive queue. (Suppressed by the 'q' option.)

count.ziph:Histogram of on node messages processed before a waiting message in Mercury's receive queue. (Suppressed by the 'q' option.)

error.check: Sometimes produced for messages with larger than 60 zip ahead messages.

stderr: Lots of messages. For example the slowest GVT message so far is outputed to stderr. (These GVT messages are suppressed by the 'g' option.)

mmstat: One line for each off node messages in a tab delimited format: "number of hops", "number of packets", "twhg time", "hghg time", "hgtw time", "twtw time", "queh number", "dlen number", and "ziph number". (Suppressed by 'm' option.)

OPTIONS:

The options all suppress the creation of certain files. Mmstat is by far the only large file. The option 'q' will save time, but should only be used with the 'm' option as it will make some of mmstat's fields wrong. The 'h', 'c' and 'g' options will not save any noticable time.

SEE ALSO:

hgram, history, mstat, tics.

BUGS:

The tics file needs to be updated regularly. If you are getting h_insert errors, then chances are you need to update "tics". (The file tics contains correcting values for the clocks on the different nodes of the hypercube.)

Man Page

tics

The file tics contains correcting values for the clocks on the different nodes of the hypercube. The file tics is created by linking timetest.o with the Time Warp code and using a config file with one line: "*timetest". (The output is needs to be redirected.)

SEE ALSO:
     exstat, hc, m3plot, msgstat.
FILES:
     timetest.c