

XMEMCK(8)

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SYNOPSIS	1
2	DESCRIPTION	2
3	OPTIONS	3
4	EXAMPLE RUN	4
5	EXIT STATUS	5
6	BUGS	6
7	SEE ALSO	7
8	COPYING	8

Chapter 1

SYNOPSIS

xmemck [*OPTIONS*]

Chapter 2

DESCRIPTION

xmemck tool is used internally by Enduro/X testing suite. The tool is started before running XATMI level tests. The tool scans the process memory usage over the time. When scanned processes are finished, tool takes some range of the Virtual Memory (VSZ) and Resident Memory (RSS) values of the snapshots (made at certain intervals) and compares average of the first halve with average of the second halve. If the increase in second halve are bigger then defined percentage, then memory leak is reported for the binary.

The monitored binaries are selected by regular expressions passed to the command line. Also for different expressions different percentage settings can be used with different monitoring parameters. The command line arguments might be repeated thus at every *-m* the new monitoring set is produced. For monitoring processes are selected by *ps -ef* output matched against regexp.

Also it is possible to disable some keywords by regular expressions too. The **xmemck** can be terminated with SIGTERM.

Chapter 3

OPTIONS

[-p] *PERIOD*

Number of seconds to scan for new processes and read existing processes VSZ and RSS. Default is **1** second.

[-d] *INCREASE_DELTA_ALLOW*

This is percentage to allow to increase between to halve average memory usage. Default is **5** percent.

[-s] *HALVE_START_PERCENT*

This is number of percents taken from the total pool of process running snapshots, to be marked as start of the first halve. For example if process have run for 180 seconds, $-p = 1$ then we have 180 snapshots of data. If $-s$ is set to **40**, then first halve is calculated from 72 sec. Default is **40**.

[-t] *HALVE_STOP_PERCENT*

This is ending percentage of the snapshots used for average halving. For example if runtime was 180 sec, start ($-s$) was **40** and stop ($-t$) is **90**, then we get following setup: average counting from 72 sec till 162 sec. Total 90 snapshots, first halve 72..117 and ending 118..162.

[-n] *NEGATIVE_REGEX_MASK*

If set, then when **xmemck** is selecting the processes for monitoring the given *ps -ef* line must not match the $-n$ parameter. If matches, then process is not monitored.

[-v] *MIN_VALUES*

Minimum number of values to use in one halve. If the number of snapshots is less than this number for the halve, then memory leak detection is not performed on this process.

-m *REGEX_MASK*

Regex mask used for process selection from *ps -ef* output. If line matches the given expression, then PID is extracted and binary is monitored at $-p$ interval for VSZ and RSS.

Chapter 4

EXAMPLE RUN

Sample run of the binary:

```
$ xmemck -v20 -d30 -s60 -t95 -n 'atmiunit1|tpbridge|tmsrv' -m atmi -d70 -m tpbridge -m ↵  
tmsrv 2>./memck.log 1>./memck.out &
```

The given example monitors following:

- (-v20) Min values 20
- (-d30) Allow delta 30%
- (-s60) Start at 60% of the runtime snapshots
- (-t95) Stop calculating averages of the 95% of snapshots
- (-n *atmiunit1|tpbridge|tmsrv*) Do not match *atmiunit1*, *tpbridge* or *tmsrv*
- (-m atmi) Match "atmi"
- (-d90) Change delta to 70 for next match
- (-m tpbridge) Monitor process named **tpbridge**
- (-m tmsrv) Monitor process named **tmsrv**

When leak is detected, the following infos is printed to stdout:

```
>>> LEAK pid=8605! rss: 21766 -> 32712 (+33.461727%), vsz 25412 -> 36337 (+30.065773%): [ ↵  
mvtolin 8605 7319 0 22:05 pts/17 00:00:00 ./a.out]
```

Chapter 5

EXIT STATUS

0

Success

1

Failure

Chapter 6

BUGS

Report bugs to support@mavimax.com.

Chapter 7

SEE ALSO

xadmin(8)

Chapter 8

COPYING

© Mavimax, Ltd.