

Speed Up Your System With Preload On Ubuntu 9.04

Version 1.0

Author: Falko Timme <ft [at] falkotimme [dot] com>

Last edited 07/06/2009

This tutorial explains how you can install [preload](#) on Ubuntu 9.04. preload is an adaptive readahead daemon. It monitors applications that users run, and by analyzing this data, predicts what applications users might run, and fetches those binaries and their dependencies into memory for faster startup times.

This document comes without warranty of any kind! I do not issue any guarantee that this will work for you!

Installing Preload

To install preload, open a terminal (Applications > Accessories > Terminal):



Then type

```
sudo aptitude install preload
```

to install preload.

That's it already. Preload will run silently in the background. If you want to change the way preload behaves, you can modify its configuration file, `/etc/preload.conf`. Here's the file with the default configuration:

```
sudo gedit /etc/preload.conf
```

```
[model]

# cycle:
#
# This is the quantum of time for preload. Preload performs
# data gathering and predictions every cycle. Use an even
# number.
```

```
#
# Note: Setting this parameter too low will may reduce the
# system performance and stability.
#
# unit: seconds
# default: 20
#
cycle = 20

# usecorrelation:
#
# Whether correlation coefficient should be used in the prediction
# algorithm. There are arguments both for and against using it.
# Currently it's believed that using it results in more accurate
# prediction. The option may be removed in the future.
#
# default: true
usecorrelation = true

# minsize:
#
# Minimum sum of the length of maps of the process for
# preload to consider tracking the application.
#
# Note: Setting this parameter too high will make preload less
# effective, while setting it too low will make it eat
# quadratically more resources, as it tracks more processes.
#
# unit: bytes
# default: 2000000
#
minsize = 2000000

#
# The following control how much memory preload is allowed to use
# for preloading in each cycle. All values are percentages and are
# clamped to -100 to 100.
#
# The total memory preload uses for prefetching is then computed using
# the following formulae:
#
#           max (0, TOTAL * memtotal + FREE * memfree) + CHACED * memcached
# where TOTAL, FREE, and CHACED are the respective values read at
# runtime from /proc/meminfo.
#

# memtotal: percentage of total memory
#
# unit: signed_integer_percent
# default: -10
#
memtotal = -10

# memfree: percentage of free memory
#
# unit: signed_integer_percent
# default: 100
#
memfree = 100

# memcached: percentage of cached memory
#
# unit: signed_integer_percent
```

```
# default: 30
#
memcached = 30

#####

[system]

# doscan:
#
# Whether preload should monitor running processes and update its
# model state. Normally you do want that, that's all preload is
# about, but you may temporarily want to turn it off for various
# reasons like testing and only make predictions. Note that if
# scanning is off, predictions are made based on whatever processes
# have been running when preload started and the list of running
# processes is not updated at all.
#
# default: true
doscan = true

# dopredict:
#
# Whether preload should make prediction and prefetch anything off
# the disk. Quite like doscan, you normally want that, that's the
# other half of what preload is about, but you may temporarily want
# to turn it off, to only train the model for example. Note that
# this allows you to turn scan/predict or or off on the fly, by
# modifying the config file and signalling the daemon.
#
# default: true
dopredict = true

# autosave:
#
# Preload will automatically save the state to disk every
# autosave period. This is only relevant if doscan is set to true.
# Note that some janitory work on the model, like removing entries
# for files that no longer exist happen at state save time. So,
# turning off autosave completely is not advised.
#
# unit: seconds
# default: 3600
#
autosave = 3600

# mapprefix:
#
# A list of path prefixes that controll which mapped file are to
# be considered by preload and which not. The list items are
# separated by semicolons. Matching will be stopped as soon as
# the first item is matched. For each item, if item appears at
# the beginning of the path of the file, then a match occurs, and
# the file is accepted. If on the other hand, the item has a
# exclamation mark as its first character, then the rest of the
# item is considered, and if a match happens, the file is rejected.
# For example a value of !/lib/modules;/ means that every file other
# than those in /lib/modules should be accepted. In this case, the
# trailing item can be removed, since if no match occurs, the file is
# accepted. It's advised to make sure /dev is rejected, since
# preload doesn't special-handle device files internally.
#
```

```
# default: (empty list, accept all)
mapprefix = /usr;/lib;!/

# exeprefix:
#
# The syntax for this is exactly the same as for mapprefix. The only
# difference is that this is used to accept or reject binary executable
# files instead of maps.
#
# default: (empty list, accept all)
exeprefix = !/usr/sbin;!/usr/local/sbin;/usr;!/
```

If you change the configuration, you need to restart preload:

```
sudo /etc/init.d/preload restart
```

Links

- preload: <http://sourceforge.net/projects/preload/>
- Ubuntu: <http://www.ubuntu.com/>