

22.10.1 Exkurs: Server Memcached – Installation und Test

Wenn Sie die Komponente *gb.memcached* testen wollen – die Ihnen einen Memcached-Client zur Verfügung stellt – dann benötigen Sie den Memcached-Server (MCS) auf Ihrem System.

22.10.1.1 Installation des Servers Memcached

Eine ausführliche Beschreibung des Protokolls des Servers *Memcached* finden Sie auf den folgenden Webseiten:

<https://github.com/memcached/memcached/blob/master/doc/protocol.txt>

Der Server *Memcached* kann über die Anwendungsverwaltung installiert werden oder Sie geben in einem Terminal folgende Zeilen ein:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install memcached
```

22.10.1.2 Memcached-Server-Status und Memcached-Server-Steuerung

So ermitteln Sie den Status des Server Memcached, da der Server Memcached nach der Installation sofort gestartet wird und auf dem (Standard-)Port 11211 lauscht:

```
$ service memcached status
oder
$ /etc/init.d/memcached status
```

Ausgabe in einer Konsole:

```
hans@mint20:~$ service memcached status
● memcached.service - memcached daemon
   Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-03-23 11:53:29 CET; 33min ago
     Docs: man:memcached(1)
  Main PID: 4900 (memcached)
    Tasks: 10 (limit: 18941)
   Memory: 2.0M
   CGroup: /system.slice/memcached.service
           └─4900 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.1 -P
             /var/run/memcached/memcached.pid

Mär 23 11:53:29 mint20 systemd[1]: Started memcached daemon.
```

Allgemein gilt für die Steuerung:

```
$ /etc/init.d/memcached {start|stop|restart|force-reload|status}
oder
$ service memcached {start|stop|restart|force-reload|status}
```

- Den Status können Sie stets ohne Root-Rechte abrufen.
- Bei allen anderen Parametern werden Sie zur Eingabe des Benutzer-Passworts aufgefordert.

Wenn Sie ständig mit dem Memcached-Server arbeiten, dann lohnt das Anlegen von Startern auf dem Desktop für ausgewählte Aktionen wie zum Beispiel das Starten oder das Stoppen.

Die Standard-Konfiguration muss im Normalfall nicht geändert werden.

Die Konfigurationsdatei finden Sie hier:

`/etc/memcached.conf`

Log-Dateien finden Sie unter:

`/var/log/memcached.log`

22.10.2 Memcached-Server – 1. Test

Für einen ersten Test, der Ihnen auch die grundlegende Arbeitsweise von MC-Server und MC-Client zeigen soll, können Sie das Programm *telnet* als Memcached-Client einsetzen – mit der IP-Adresse 127.0.0.1 und dem Port 11211.

Das folgende Konzept soll umgesetzt werden:

- Zum Memcached-Server verbinden
- Daten speichern: Unter dem Schlüssel **TESTKEY** soll das Wort **MINISINI** gespeichert werden
- Daten auslesen
- Existierende Daten überschreiben: **MINISINI** wird mit **GIANLUIGI** überschrieben
- Daten löschen zu einem bestimmten Schlüssel
- Ausgabe von Server-Statistiken
- Server-Cache komplett löschen
- Verbindung zum Memcached-Server trennen

Die Eingaben sind in der Farbe rot markiert, Memcached-Schlüsselworte in fett und die Ausgaben in der Farbe blau:

```
# telnet hostname/ip port
hans@mint20:~$ telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
# version
version
VERSION 1.5.22
# set key_name meta_data expiry_time_in_seconds length_in_bytes
set TESTKEY 0 100 8 # + ENTER
MINISINI
STORED
# get key_name
get TESTKEY
VALUE TESTKEY 0 8
MINISINI
END
# replace key_name meta_data expiry_time_in_seconds length_in_bytes
replace TESTKEY 0 100 9
GIANLUIGI
STORED
# delete key_name
delete TESTKEY
DELETED
# stats
stats
STAT pid 4900
STAT uptime 9032
STAT time 1616505839
STAT version 1.5.22
...
STAT direct_reclaims 0
STAT lru_bumps_dropped 0
END
# flush_all
flush_all
OK
# quit
quit
Connection closed by foreign host.
hans@mint20:~$
```

Übersicht zu ausgewählten Meldungen des MC-Servers:

- **STORED** zeigt den Erfolg beim Speichern von Daten an.
- **ERROR** zeigt einen Fehler beim Speichern von Daten oder eine falsche Syntax an.
- **EXISTS** zeigt an, dass jemand die CAS-Daten seit dem letzten Abruf geändert hat.
- **NOT_FOUND** zeigt an, dass der Schlüssel im Memcached-Server nicht vorhanden ist.
- **DELETED** zeigt das erfolgreiche Löschen von Daten zum angegebenen Schlüssel an.

22.10.3 Memcached-Server – 2. Test

Seine Stärken spielt der Server Memcached erst dann aus, wenn Sie die zu speichernden Daten in einer modifizierten Form abspeichern.

- Zum Memcached-Server verbinden
- Daten speichern
- Daten auslesen
- Existierende Daten überschreiben
- Daten löschen zu einem bestimmten Schlüssel
- Ausgabe von Server-Statistiken
- Server-Cache komplett löschen
- Verbindung zum Server trennen

```
# telnet hostname/ip port
hans@mint20:~$ telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
# set key_name meta_data expiry_time_in_seconds length_in_bytes
set TestKey 0 100 6 # ENTER
Gambas
STORED
# get key_name
get TestKey
VALUE TestKey 0 6
Gambas
END
# replace key_name meta_data expiry_time_in_seconds length_in_bytes
replace TestKey 0 100 13
Gambas 3.15.2
STORED
# delete key_name
delete TestKey
DELETED
# stats
stats
STAT pid 4900
STAT uptime 9032
STAT time 1616505839
STAT version 1.5.22
...
STAT direct_reclaims 0
STAT lru_bumps_dropped 0
END
# flush_all
flush_all
OK
# quit
quit
Connection closed by foreign host.
hans@mint20:~$
```

Ob Memcached als Caching-Lösung die beste Option ist, hängt von den Ansprüchen und der Komplexität der jeweiligen Anwendung ab.

Besonders sinnvoll ist das temporäre Speichersystem zum Beispiel für Webanwendungen mit hohem Traffic und Seiten, die auf riesige Datenbanken zugreifen.