LU14b - LogFile Management

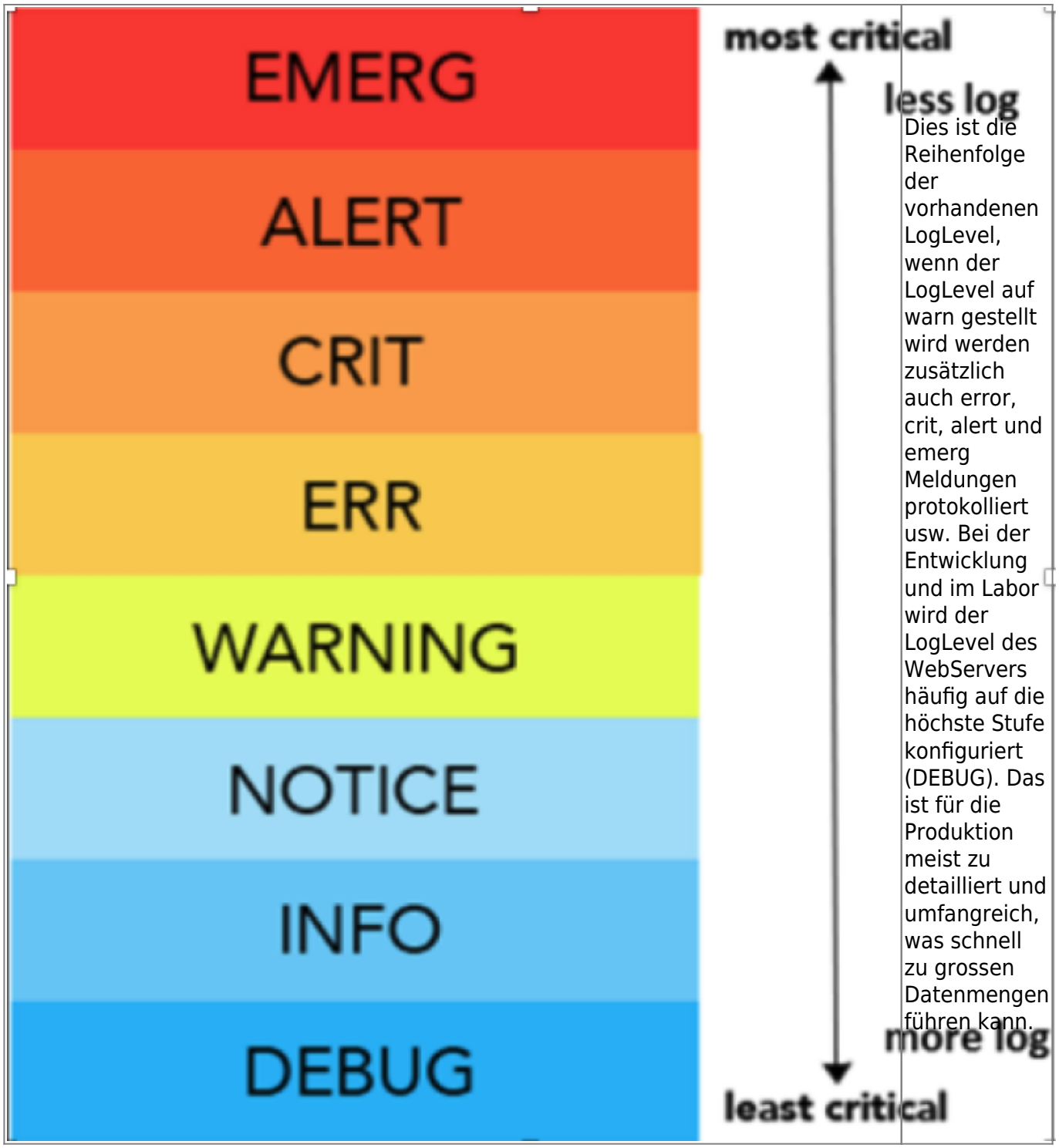
Lernziele

- Level der Protokollierung von Logfiles nennen und erklären können.
- Das Rotationsprinzip bei Logfiles erklären können.
- Elemente eines Logfiles nennen und darlegen können.

Konfiguration des LogLevels

Der LogLevel sagt aus welchen Informationen in der ErrorLog-Datei gespeichert werden. Dabei gibt es die folgenden Level mit ihrer Bedeutung.

Level	**Beschreibung**
emerg	Dies protokolliert nur Abstürze oder Ausfälle des Servers.
alert	Dies protokolliert Ereignisse, nach denen der Serveradministrator dringend eingreifen sollte.
crit	kritische Situationen die den Betrieb des Servers beeinträchtigen können.
error	Fehlermeldungen die Beachtung finden sollten.
warn	einfache Warnmeldungen
notice	Notizen des Webservers die immer protokolliert werden. Zum Beispiel das Starten und Stoppen des Servers.
info	reine Informationsmeldungen die keine Fehler sind
debug	allgemeine Meldungen des Servers werden in dieser Stufe protokolliert.



Dies ist die Reihenfolge der vorhandenen LogLevel, wenn der LogLevel auf warn gestellt wird werden zusätzlich auch error, crit, alert und emerg Meldungen protokolliert usw. Bei der Entwicklung und im Labor wird der LogLevel des WebServers häufig auf die höchste Stufe konfiguriert (DEBUG). Das ist für die Produktion meist zu detailliert und umfangreich, was schnell zu grossen Datenmengen führen kann.

LogFile Management

Werden Pfade für die LogFiles nicht explizit angegeben, so werden diese normalerweise im Unterverzeichnis /logs des Installationsverzeichnisses vom WebServer oder unter /var/log abgelegt. Da LogFiles sehr schnell eine beachtliche Grösse erreichen können, droht ein Überlaufen der entsprechenden Partition. Dies kann zu ernsthaften Störungen des WebServer Betriebs führen. Das bedeutet, dass die LogFiles regelmässig verschoben oder gelöscht werden müssen.

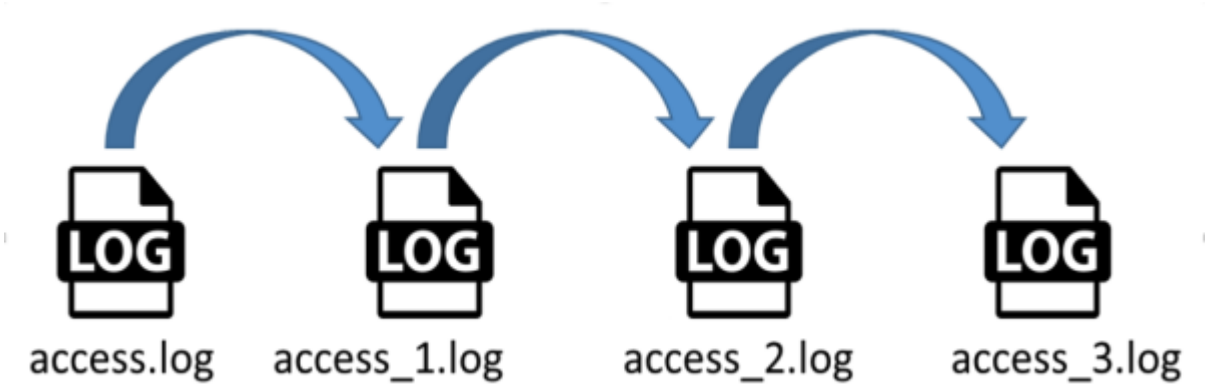
Logrotate

Es ist nicht möglich, während dem ein WebServer läuft die LogFile zu verschieben, stattdessen muss der Server neu gestartet werden, um die alten LogFiles durch neue auszutauschen. Dies kann manuell durch Verschieben der Dateien und dann Soft-Restarting durchgeführt werden, so dass die Einträge in die neuen LogFiles geschrieben werden können.

Fürr einen Apache WebServer könnte das wie folgt aussehen:

```
mv access_log access_1.log
mv error_log error_1.log
apachectl graceful
```

Durch diese Kommandos werden die LogFiles verschoben und der WebServer neu gestartet. Anschliessend werden 600 Sekunden gewartet, sodass Apache die alten LogFiles weiterhin verwenden kann, um die Protokollierung von laufenden Anfragen abzuschliessen. Während dieser Zeit werden neue Anfragen in die neuen, aktualisierten LogFiles protokolliert.



Einfacher geht es durch den Einsatz eines Programmes, wie logrotate3. Die Konfiguration von logrotate erfolgt unter Linux in der Datei /etc/logrotate.conf. Die Syntax ist relativ einfach. Das zu „rotierende“ LogFile wird eintragen und in den darauffolgenden geschweiften Klammern festlegen, wie das LogFile behandelt werden sollen.

Beispiel von logrotate.conf

/var/www/access_log {

```
weekly      # Wöchentlich rotieren lassen
compress        # Gesicherte Log-Dateien komprimieren
rotate 5        # Max. 5 mal rotieren (5 Backups erstellen)
postrotate      # Nach dem Rotieren werden Skripte ausgeführt
    /bin/killall -HUP apache2
                    # Neustart von Apache-Prozessen,
                    # damit neue Log-Datei genutzt wird
endscript           # Ende von postrotate
```

Damit logrotate regelmässig ausgeführt wird, kann ein *cron job* erstellt werden (auch CRONTAB = im Prinzip der Fahrplan mit Jobs und Zeiten, zu denen diese ausgeführt werden sollen). Mit der folgenden Konfigruation wird //logrotate täglich ausgeführt:

```
/usr/sbin/logrotate -s /etc/cron.daily/logrotate.status
/etc/cron.daily/logrotate.conf
```

Volkan Demir

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m183/learningunits/lu08/02?rev=1765181985

Last update: **2025/12/08 09:19**