

UNIX-Dämonen mit Perl

Will man ein Perlscript dauerhaft im Hintergrund laufen lassen, muss es von der Konsole getrennt und in den Hintergrund geschickt werden. Folgendes Beispiel zeigt wie das geht:

```
#!/usr/bin/perl
#
# 2008 Oliver Voelker <info(at)ovtec.it>
#
use strict;
use warnings;
use POSIX qw(setsid);
my $logfile = "/var/log/mein-logfile";      # logfile
#my $logfile = "/dev/null";                # aka don't log

sub daemonize {
    # change to /-directory, so that the daemon doesn't block any mountpoints
    chdir("/") or die("Can't chdir to /: $!");
    # redirect STDIN, SDOUT and STDERR
    open(STDIN, "/dev/null") or die("Can't read /dev/null: $!");
    open(STDOUT, ">>$logfile") or die("Can't write to $logfile: $!");
    open(STDERR, ">>$logfile") or die("Can't write to $logfile: $!");
    # fork a child and exit() the parent (when the child was created successfully) to
    disassociate from the login shell
    # also setsid would not work without that
    defined(my $pid = fork) or die("Can't fork: $!");
    exit if($pid);
    # run this program in a new session
    setsid or die("Can't start a new session: $!");
    # set the creation mode for new files
    umask(0);
}

# flush the buffer
$| = 1;

# daemonize the program
&daemonize;
```

```
my $count = 0;
# now do something in a loop
while(1) {
    $count++;
    print "I've done this $count times already!\n";
    sleep(5);
}
```

Wenn man dieses Programm jetzt startet, landet man sofort wieder auf dem Prompt, da sich das Programm sofort von der Shell trennt und in den Hintergrund verabschiedet. Die korrekte Funktion kann man jetzt z.B. über das Logfile (/var/log/mein-logfile) oder via „strace -p <pid“ prüfen.

```
[root@bla bin]# tail -f /var/log/mein-logfile
I've done this 36 times already!
I've done this 37 times already!
I've done this 38 times already!
I've done this 39 times already!
I've done this 40 times already!
I've done this 41 times already!
I've done this 42 times already!
```

```
[root@bla bin]# strace -p 11786
Process 11786 attached - interrupt to quit
restart_syscall(<... resuming interrupted call ...>) = 0
time(NULL) = 1227252023
write(1, "I've done this 11 times already!"..., 33) = 33
time(NULL) = 1227252023
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigaction(SIGCHLD, NULL, {SIG_DFL}, 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
nanosleep({5, 0}, {5, 0}) = 0
time(NULL) = 1227252028
write(1, "I've done this 12 times already!"..., 33) = 33
time(NULL) = 1227252028
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigaction(SIGCHLD, NULL, {SIG_DFL}, 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
nanosleep({5, 0}, {5, 0}) = 0
time(NULL) = 1227252033
write(1, "I've done this 13 times already!"..., 33) = 33
time(NULL) = 1227252033
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
```

```
rt_sigaction(SIGCHLD, NULL, {SIG_DFL}, 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
nanosleep({5, 0}, {5, 0})                = 0
```

Revision #1

Created 2021-07-27 09:59:29 UTC by magenbrot

Updated 2021-07-27 10:00:34 UTC by magenbrot