

**NAME**

akfavatar-term – Modul zum Ausführen von Terminal-basierten Programmen in AKFAvatar

**SYNTAX**

term = require "akfavatar-term"

**Hinweis:** Man muss eine globale Variable verwenden, wenn man die APC-Schnittstelle verwenden will.

**BESCHREIBUNG**

Dieses Modul kann man verwenden um allgemeine Terminal-basierte Programme innerhalb von AKFAvatar laufen zu lassen. Die Terminal-Emulation emuliert eine Linux-Textkonsole (nicht nur auf Linux-basierten Systemen).

Es kann auch dafür verwendet werden, um spezifische Programme für AKFAvatar zu schreiben, in Programmiersprachen, die ansonsten nicht unterstützt werden. Es gibt ein paar Möglichkeiten mit denen das ausführende Skript und das auszuführende Programm miteinander kommunizieren können.

**Achtung:** Dieses Modul steht nicht für allen Plattformen zur Verfügung! Insbesondere nicht für Windows.

**term.startdir([Verzeichnis])**

Legt das Anfangsverzeichnis für den nächsten **term.execute()**-Befehl fest. Wenn kein *Verzeichnis* angegeben wird, wird die vorherige Einstellung dadurch gelöscht.

**Hinweis:** Das Arbeitsverzeichnis des ausführenden Skriptes und des ausgeführten Programmes sind voneinander unabhängig!

**term.homedir()**

Setzt das Anfangsverzeichnis für den nächsten **term.execute()**-Befehl auf das Heimatverzeichnis des Benutzers.

**Hinweis:** Das Arbeitsverzeichnis des ausführenden Skriptes und des ausgeführten Programmes sind voneinander unabhängig!

**term.color(true|false)**

Schaltet die Farbunterstützung ein oder aus. Der Terminal-Typ mit Farben ist „linux“, ohne Farben „linux-m“.

**term.setenv(Variable, Wert)**

Setzt die angegebene Umgebungsvariable auf den angegebenen Wert.

Das hat auch Einfluss auf Programme, die mit **os.execute()** oder **io.popen()** ausgeführt werden.

**term.unsetenv(Variable)**

Löscht die angegebene Umgebungsvariable.

Das hat auch Einfluss auf Programme, die mit **os.execute()** oder **io.popen()** ausgeführt werden.

**term.execute([Programm [, Argumente]])**

Führt das angegebene *Programm* mit den angegebenen *Argumenten* in der Terminal-Emulation aus. Wenn kein *Programm* angegeben wird, wird die Standard-Shell ausgeführt.

**APC-Schnittstelle**

Die „Application Program Command“ (APC) Schnittstelle ist eine Methode mit der das ausgeführte Programm mit dem Lua-Interpreter des ausführenden Programmes kommunizieren kann. Um dies zu verwenden sendet das ausgeführte Programm Escape-Sequenzen der Art, wie ESC + „\_“, gefolgt vom Befehl, abgeschlossen mit ESC + „\“. Die Ausgabe muss eventuell noch einen Flush bekommen, um einen sofortigen Effekt zu erzielen.

Zum Beispiel in C:

```
static void
APC (const char *s)
{
    fprintf (stdout, "\033_ %s\033\\", s);
    fflush (stdout);
}
```

Die Befehle können Lua-Funktionsaufrufe sein, insbesondere die „avt.“-Befehle von „lua-akfavatar“ (siehe „lua-akfavatar-ref“).

Bis zu 1024 Zeichen können gesendet werden. Das bedeutet, man kann keinen komplizierten Code vom ausgeführten Programm senden. Allerdings kann man im Lua-Skript, das das Programm aufruft globale Funktionen definieren, die dann auch mit der APC-Schnittstelle aufrufbar sind.

Als Konvention sollte das aufrufende Skript eine Umgebungsvariable namens „APC“ definieren, die die verfügbaren Funktionen beschreibt.

Zum Beispiel:

```
term.setenv("APC", _VERSION .. " ", lua-akfavatar")
```

**term.send(*String*)**

Sendet einen String an das ausgeführte Programm, so als ob es von der Tastatur eingegeben worden wäre. Das kann man nur von der APC-Schnittstelle (siehe oben) verwenden.

Wenn das ausgeführte Programm zeilenorientiert ist, sollte der String mit einem „\r“ (wie Return) abgeschlossen werden.

**term.decide(*String1* [,*String2*])**

Zeige Plus-/Minus-Knöpfe und abhängig von dieser Auswahl sende *String1* oder *String2* an das ausgeführte Programm, so als ob es von Tastatur eingegeben worden wäre. Das kann man nur von der APC-Schnittstelle (siehe oben) verwenden.

Wenn das ausgeführte Programm zeilenorientiert ist, sollten beide Strings angegeben werden und beide mit einem „\r“ (wie Return) abgeschlossen werden.

**SIEHE AUCH**

**lua-akfavatar(1) lua(1) lua-akfavatar-ref(3) akfavatar-graphic(3) akfavatar.utf8(3)**