

# SA-MP

Installation von SA-MP Servern und weitere nützliche Dinge

- [SA-MP Server installieren](#)
- [Gamemodes auf einem Linux System kompilieren](#)

# SA-MP Server installieren

## Host aktualisieren und benötigte Pakete installieren

```
sudo apt update && sudo apt upgrade -y  
sudo apt install lib32stdc++6 lib32z1 screen wget
```

## Benutzerkonto anlegen

```
sudo adduser sampserver --gecos ""
```

Passwort nach Aufforderung eingeben, mit Enter quittieren und nochmal eingeben (zur Bestätigung)

## Serverdateien herunterladen

Hierzu wechseln wir in den eben erstellten Benutzer

```
sudo su - sampserver
```

und laden dann die Serverdateien herunter

```
wget -q -O - https://files.sa-mp.de/server/0.3.X/0.3.7/linux/samp037svr_R3.tar.gz | tar xvfz -
```

Die Serverdateien wurden automatisch entpackt und liegen nun im Verzeichnis **samp03**

## Abschließende Schritte

Um den Server starten zu können muss das RCON Passwort geändert werden

```
cd samp03/  
sed -i 's/changeme/DEIN_NEUES_RCON_PASSWORT/g' server.cfg
```

Nun kann der Server gestartet werden

```
./samp03svr
```

# Server im Hintergrund betreiben

Würde man nun das Terminalfenster (bspw. PuTTY) schließen würde auch unser SA-MP Server beendet werden. Um dies zu verhindern kann man sich dieses kleinen Skriptes bedienen

```
#!/usr/bin/env bash

start() {
    echo "Starte den SA-MP Server"
    screen -S samp_server -m -d sh -c './samp03svr'
    sleep 2
    echo "SA-MP Server wurde mit der PID $(ps fax | grep samp03svr | grep -v grep | head -1 | awk '{print $1}')
```

gestartet"

```
    echo "Status: [$(tput setaf 2)OK$(tput sgr0)]"
}

stop() {
    echo "Stoppe den SA-MP Server"
    screen -X -S samp_server quit
    sleep 2
    echo "Status: [$(tput setaf 2)OK$(tput sgr0)]"
}

case "$1" in
    start) start ;;
    stop) stop;;
    *) echo "Auswahl: $0 start|stop" >&2
       exit 1
       ;;
esac
```

Den gesamten Code kopieren und in eine neue Datei einfügen

```
nano control.sh
--- Code einfügen, Editor mit CTRL-X -> Y -> ENTER beenden ---
chmod +x control.sh
```

# Gamemodes auf einem Linux System kompilieren

## Compiler herunterladen

Wir finden den Compiler auf GitHub, einfach den aktuellsten Release herunterladen

```
mkdir -p /tmp/pawn && cc /tmp/pawn
wget -q -O - https://github.com/pawn-lang/compiler/releases/download/v3.10.10/pawnc-3.10.10-linux.tar.gz | tar
xvzf -
chmod a+x /tmp/pawn/pawnc-3.10.10-linux/bin/pawnc
```

Wir finden die Compiler-Executable dann im Verzeichnis `/tmp/pawn/pawnc-3.10.10-linux/bin/pawnc`

## Library installieren

Um den Compiler ausführen zu können muss eine mitgelieferte Library-Datei in das passende Verzeichnis auf dem System kopiert werden

```
cp /tmp/pawn/pawnc-3.10.10-linux/lib/libpawnc.so /usr/lib
```

## Compiler installieren

Installieren ist vielleicht das falsche Wort, wir kopieren einfach die Binary in ein Verzeichnis unserer Wahl, z.b.

```
mkdir ~/pawn-compiler
cp /tmp/pawn/pawnc-3.10.10-linux/bin/pawnc ~/pawn-compiler/
```

## Compiler ausführen und .amx Datei erzeugen

```
cd ~/dein-gamemode/
~/pawn-compiler/pawnc "-iinclude" "gamemodes/gamemode.pwn" "-;+" "-v2" "-d3" "-Z+" "-(+"
```

Wichtig ist hierbei dass dein Gamemode (die .pwn Datei) sich im Verzeichnis *gamemodes* sowie alle benötigten Includes sich im Verzeichnis *include* befinden und nicht wie vom Windows PC gewohnt im Ordner *pawno/includes*