

## Workshop ‚Plattformen für eingebettete Systeme‘

### Aufgabenblatt 2

Daten können nicht nur über das Einhängen von Datenträgern verfügbar gemacht werden. Das Network File System (NFS) bietet auch die Möglichkeit, Verzeichnisse über das Netzwerk freizugeben und auf NFS-Clients verfügbar zu machen.

#### Aufgabe 1:

Legen Sie auf dem *Host* in Ihrem Home-Verzeichnis ein neues Verzeichnis namens `mobile-project` an und wechseln Sie in dieses Verzeichnis. Dieses Verzeichnis wollen wir in den folgenden Aufgaben als Projektverzeichnis nutzen und es zunächst über NFS exportieren.

#### Aufgabe 2:

Installieren Sie auf dem *Host* den NFS-Server und den Client:

```
sudo apt-get install nfs-kernel-server nfs-common
```

#### Aufgabe 3:

Wir konfigurieren den NFS-Server. Exporte des NFS-Servers werden über die Datei `/etc/exports` kontrolliert:

Es gibt zahlreiche Konfigurationsmöglichkeiten. Wir beschränken uns auf eine sehr einfache Konfiguration. Die folgende Zeile erlaubt allen Clients lesenden Zugriff auf Ihr Projektverzeichnis. Tragen Sie diese Zeile in die Konfigurationsdatei ein:

```
<absoluter Pfad zum Projektverzeichnis> *(ro)
```

**Achtung:** Vermeiden Sie unbedingt Leerzeichen nach dem `*`.

#### Aufgabe 4:

Halten Sie den NFS Server an und starten Sie ihn wieder:

```
sudo service nfs-kernel-server restart
```

Wir testen jetzt die Konfiguration, indem wir das exportierte Verzeichnis lokal über NFS einhängen. Für den NFS-Client benötigen wir das Paket `nfs-common`, das wir bereits in Aufgabe 2 installiert haben.

NFS-Exporte werden ganz ähnlich wie Datenträger eingehängt. Erzeugen Sie dazu – den Unix-Konventionen folgend - zunächst den Mountpunkt `nfs` im Verzeichnis `/mnt`

Anschließend hängen Sie den NFS-Export mit dem folgenden sehr klaren Befehl in den Mountpunkt ein:

```
sudo mount 127.0.0.1:<absoluter Pfad zum Projektverzeichnis> /mnt/nfs
```

Den Pfad zum Projektverzeichnis können Sie sich dabei mit

```
echo $PWD
```

anzeigen lassen. Er gibt den Inhalt der Umgebungsvariablen `PWD` aus. Diese Variable enthält den Namen des aktuellen Verzeichnisses.

### Aufgabe 5:

Prüfen Sie, ob Sie das Verzeichnis korrekt eingehängt haben: Legen Sie dazu eine Datei namens `Readme.txt` in Ihrem Projektverzeichnis an. Ist die Datei auch am Mountpunkt sichtbar?

### Aufgabe 6:

Es ist auf Dauer lästig, nach jedem Neustart des Clients neu einzuhängen. Mit Hilfe der Datei `/etc/fstab` können Sie dafür sorgen, dass *automatisch* beim Neustart eingehängt wird. Fügen Sie zu dieser Datei die folgende Zeile hinzu:

```
127.0.0.1:<absoluter Pfad zum Projektverzeichnis> /mnt/nfs nfs ro
```

Achtung: die letzte Zeile in der Datei muss eine Leerzeile sein!

Die Bedeutung sollte klar sein.

Mit dem Befehl

```
mount -a
```

können Sie jetzt alle Dateisysteme, die in `/etc/fstab` eingetragen sind aus- und wieder einhängen. Führen Sie den Befehl aus, um zu prüfen, ob Ihr Eintrag richtig war.

### Aufgabe 7:

Seit einiger Zeit sind die beiden Pakete `avahi-daemon` und `libnss-mdns` standardmäßig in der Ubuntu-Desktop Distribution enthalten.

Damit ist folgendes möglich: Es ist auf Dauer lästig für die Verbindung die IP-Adresse zu ermitteln, wenn man etwa eine ssh-Verbindung aufbauen will. Um sich mit Hilfe eines Hostnamens anzumelden, braucht man normalerweise einen Eintrag im DNS. Für Namen im *lokalen* Netzwerk, die auf `.local` enden, geht das einfacher mit dem Apple-Protokoll Bonjour. Und genau das wird mit Hilfe der beiden Pakete bereitgestellt.

Entnehmen Sie einem der Fächer im Labor eine Materialbox mit einem Raspberry Pi. *Merken Sie sich unbedingt die Nummer Ihres Faches!* Wenn XX diese Nummer Ihrer Faches xx ist, dann ändern Sie in der Datei `/etc/hostname` den Namen `ubuntu` in `hostXX.local`. Wenn Sie also die Fachnummer 7 haben, bekommt Ihre virtuelle Maschine den Namen `host07`. Starten Sie Ihren Host neu. Ab jetzt sollte Ihr Host unter seinem neuen Namen erreichbar sein.