

# BASH script voor klankinstallatie

Benodigde software:

sox <http://sox.sourceforge.net> (in Linux: sudo apt install sox)  
soxi <http://sox.sourceforge.net/soxi.html> kan ook van pas komen.

Benodigde kennis: basis 'UNIX shell and scripting'

- <https://www.dinkum.nl/software/shell>
- <https://www.lynda.com/Bash-tutorials/Learning-Bash-Scripting/142989-2.html>

## Opdracht

Een installatiekunstenaar vraagt je om hulp. Hij heeft ongeveer 50 lange samples met omgevingsgeluid ([http://www.dinkum.nl/software/shell/assignments/sox\\_fragments.tar](http://www.dinkum.nl/software/shell/assignments/sox_fragments.tar)) die verwerkt moeten worden tot verschillende korte samples.

Er zijn een paar algemene eisen waaraan de uiteindelijke korte samples moeten voldoen:

- genormaliseerd
- fade-in en fade-out van 0.1 sec. aan begin en einde

Hij wil graag elke lange sample verwerkt hebben tot:

- 5 sec. "puur geluid"
- 4 sec. "*alsof het uit een andere ruimte komt verderop in het gebouw*"
- 2 sec. "*dat een beetje kapot klinkt, zoals scheurende gitaren*"
- 6 sec. "*nog een andere variant die totaal anders klinkt dan bovenstaande, weet jij iets?*"

Tot slot wil hij ook, indien mogelijk:

- "*één track van 1 minuut, bestaande uit een willekeurige selectie van alle resulterende samples, die telkens maar 0.1 tot 1 seconde te horen zijn.*"  
Deze mogen na elkaar klinken maar ook tegelijkertijd (max 3 samples tegelijkertijd).  
Variëren in volume mag!"

Dit is in de meeste programma's een tijdrovende klus. Gelukkig kun jij scripten en je bedenkt dat een BASH-script het hele proces kan automatiseren en daarmee veel efficiënter is. Ook handig voor in de toekomst voor je eigen werk!

## Aanpak

### Ken uw gereedschap

Begin met wat experimenten met sox zodat je de mogelijkheden en parameters leert kennen. Copy-paste af en toe bruikbare instructies naar een text-file die je later gaat gebruiken als script. Je kunt online veel voorbeelden van het gebruik van sox vinden.

### First HACK IT

Maak een script voor de 5 seconden "*puur geluid*" samples.

Gebruik eerst in dit script een 'hard coded' input file. Ook de te gebruiken waarden voor de sample- en fade lengte kun je eerst 'hard coden'.

Vervang daarna deze hard coded waarden door command-line parameters die je bij aanroepen van het script kunt meegeven. Daarna ga je pas door met de volgende scripts.

### HINTS

Gebruik de SOX effecten "trim" en "fade", en gebruik "--norm" voor je eerste script.

```
$ sox sample.wav sample_trim.wav trim 6.0 3.0
```

```
$ sox --norm=-3 sample_trim.wav sample_trim_fade.wav fade 1.0 3.0
```

Gebruik de documentatie van sox: <http://sox.sourceforge.net/sox.pdf> en probeer voorbeelden uit.

Ontwikkel je script stap voor stap en test telkens of het nog steeds goed werkt. Als je nieuwe dingen toevoegt waar je nog niet helemaal zeker van bent, zet er dan eerst eens een **echo** voor om te zien wat het **zou** uitvoeren. Op die manier kun je een hoop ellende voorkomen zoals het verwijderen van files of het onbedoeld volschrijven van je Desktop met 10000 files. Neem dit serieus, we hebben het al meegemaakt! En wees vooral terughoudend/voorzichtig met het gebruik van 'mv' en 'rm'.

Intermezzo: hoe verwijder je 10000 files van je Desktop?

### **Aanvullende voorwaarden**

- Integreer git in je workflow: zet bij elke significante wijziging je script weer in git (met add/commit)
- Laat het script alle resultaten in sub-folders zetten, dus niet alles door elkaar in dezelfde folder
- Maak het bruikbaar voor anderen en help de gebruiker: als de gebruiker te weinig of onjuiste argumenten opgeeft zal het script aangeven wat er niet klopt of een korte gebruiksaanwijzing laten zien. Zorg dat het script veilig te gebruiken is.
- Als alle argumenten goed zijn opgegeven krijgt de gebruiker te zien welke acties het script gaat uitvoeren en wordt aan de gebruiker gevraagd of het uitgevoerd mag worden
- Tijdens het uitvoeren laat het script zien met welke file hij bezig is