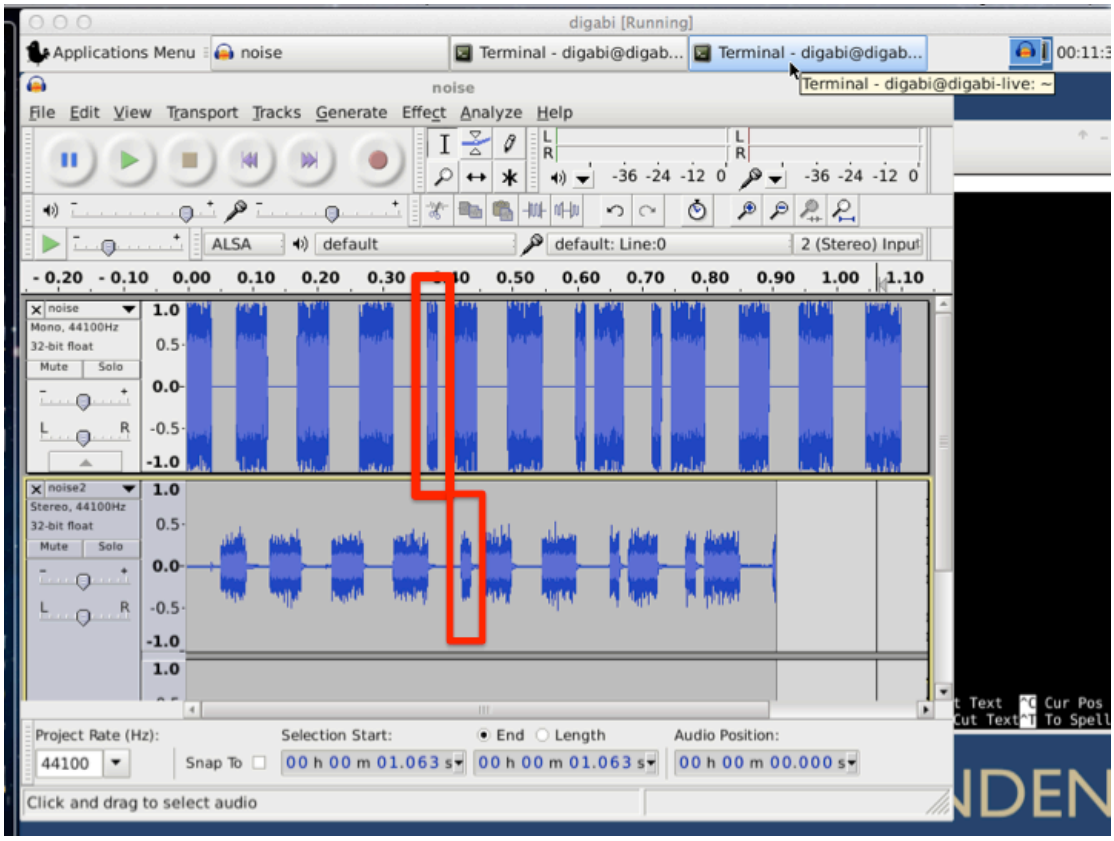
Pörisevä tietokone

morsetusta äänikortilla ja mikrofonilla



Tiivistelmä

Idea toteuttaa seuraavat vaatimukset:

1. kommunikointi toisen opiskelijan kanssa (morsetus)
2. toisen opiskelijan häirintä (keskittymistä häiritsevä ääni)

Kommunikointi tapahtuu tuottamalla tietokoneesta ääntä, johon on koodattu sanoma. Viesti vastaanotetaan tietokoneen mikrofonilla. Kantoaallon taajuus voidaan säätää ihmisen kuulokynnyksen ylä- tai alapuolelle, tai naamioida esimerkiksi napin painalluksiksi tai tuulettimen vikinäksi.

Häirintä tapahtuu tuottamalla korkeataajuista vikinää, joka aiheuttaa oaslle ihmisistä pahoinvointia ja keskittymishäiriöitä.

Kommunikointi ja häirinrtä perustuu ‘‘Advanced linux sound architecture’’ -komentojen sallimiseen peruskäyttäjillä (komennot aplay ja arecord), sekä Python-ohjelmointikieleen, jolla voidaan kirjoittaa noin 20 riviä pitkä kommunikointiohjelma. Tämä on ohjelmointitaitoiselle helppoa.

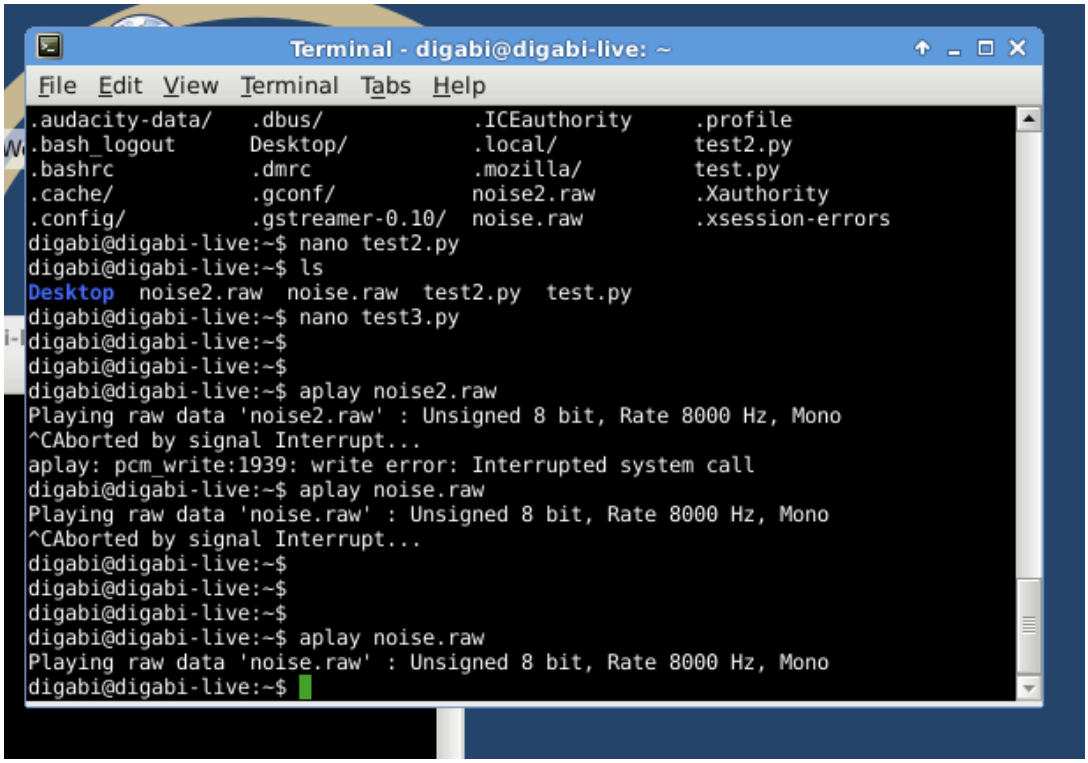
Menetelmän kuvaus

Viestien lähetys

Lähetys perustuu komentoon

aplay noise.raw

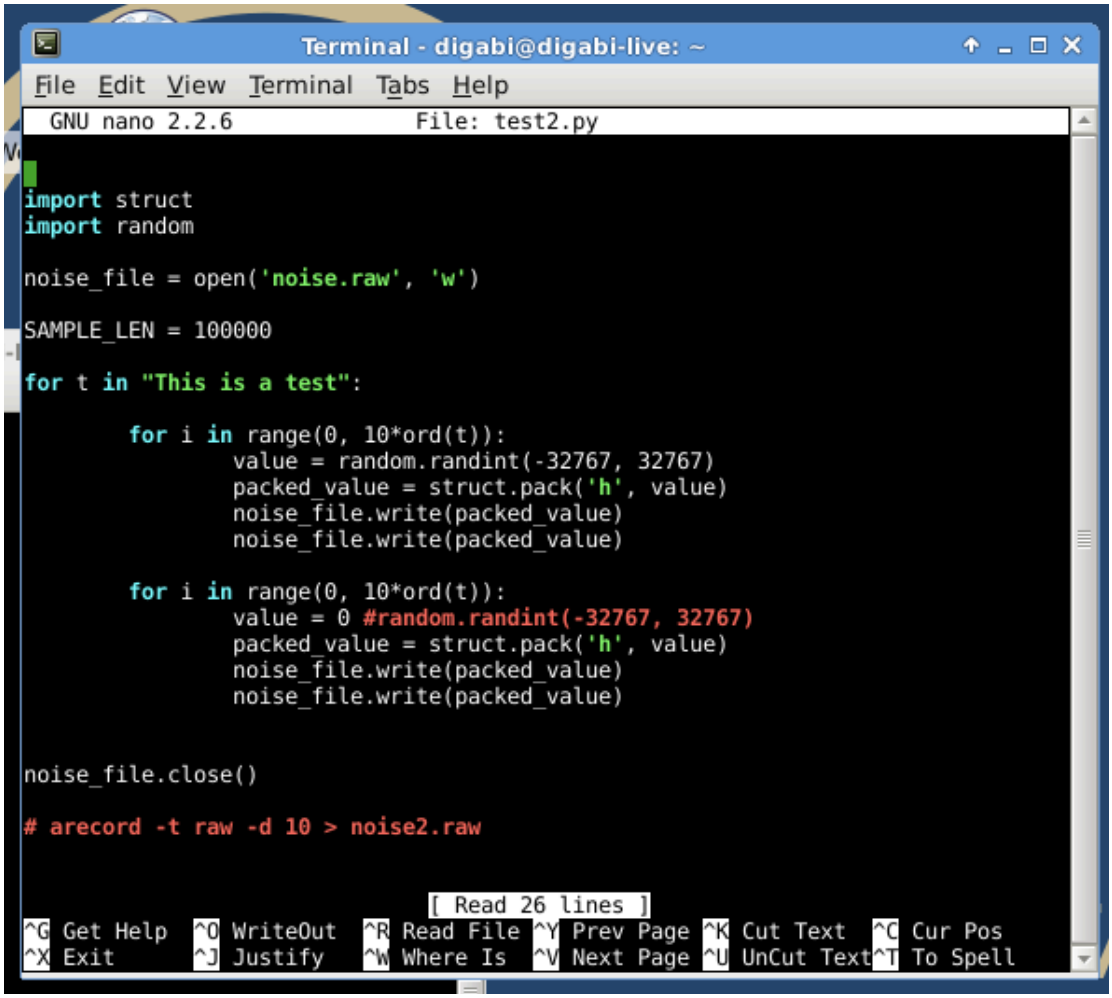
jossa *noise.raw* sisältää lähetettävän viestin. Tämä komento toimii virtualboxissa ajettavalla Digabi-levykuvalla. Komentoa ajettaessa kaiuttimista kuuluu *noise.raw*-tiedoston sisältö. Kuvassa 1 on ruutukaappaus tapahtumasta.



Kuva 1: Terminaali-ikkunassa *noise.raw*-tiedosto soitetaan äänikortissa.

Viestien luominen

Kuva 2 esittää yksinkertaisen Python-ohjelman viestien luomiseen, joka voidaan ajaa Digabi-ympäristössä. Siinä "This is a test" –lause koodataan luvuiksi (int) kirjain kirjaimelta kohdassa 10*ord(*t*), missä *t* on yksi kirjain. Sisemmät for–loopit luovat lukua vastaavan jakson, jossa on sekä kohina että hiljainen osuus. Lauseen koodaaminen tuottaa pätkittäistä kohinaa sisältävän audio-raidan, jonka jaksojen pituuksista voidaan määrittää koodatut kirjaimet yksikäsitteisesti.



Kuva 2: Ruutukaappaus havainnollistaa *noise.raw*-tiedoston luomisen Digabi Nano-editorissa.

Viestien vastaanottaminen

Viestejä vastaanotetaan komennolla

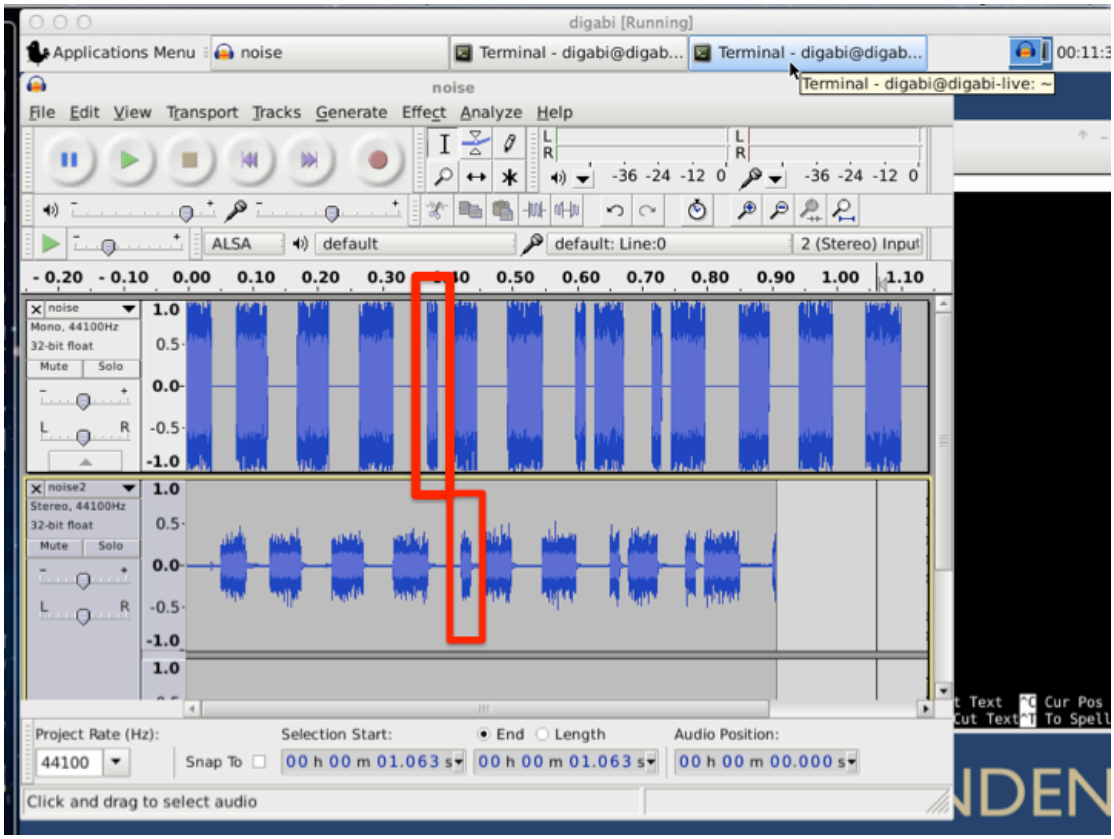
arecord –t raw –d 10 > noise2.raw

joka tallentaa tiedoston raw-muodossa. Parametri *-d 10* on äänityksen kesto ja *noise2.raw* on tiedoston nimi. Komento tallentaa tietokoneen mikrofoniin tulevan äänen tiedostoon.

Komennon saa myös tulostamaan äänen standard out:iin, jolloin Pythonin subprocess-komennolla mikrofonia voidaan tarkkailla reaaliaikaisesti.

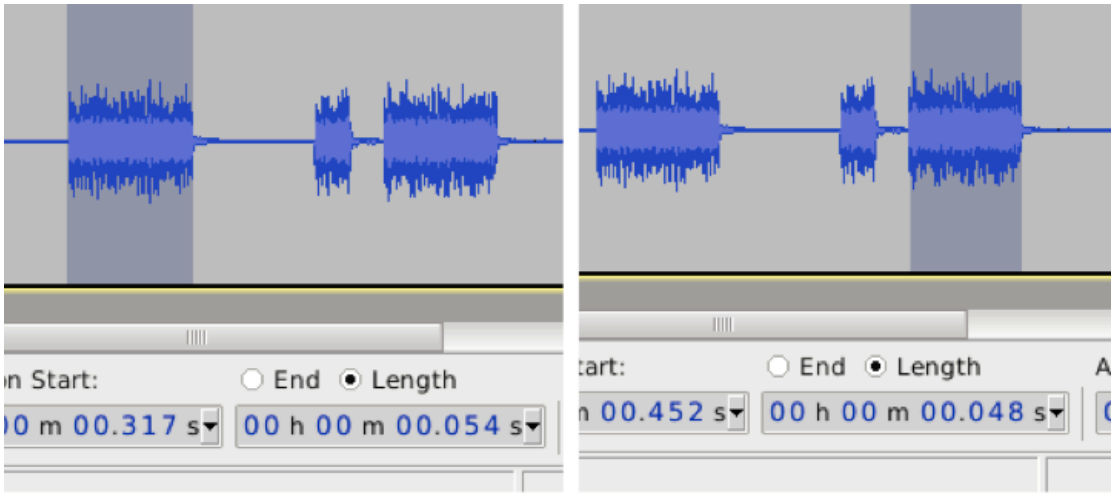
Viestien purkaminen

Viestien purkaminen tehdään periaatteessa samalla tavalla kuin niiden luominen. Pythonin struct-moduuli sisältää unpack–funktion, jolla binääridatasta saadaan floateja. Kuvat 3 ja 4 esittävät ruutukaappauksia Digabin Audacity-ohjelmasta. Kuvista selviää viestien purun periaate. Kuvassa 3 (ylempi sininen murtoviiva) on lähetetty tiedosto *noise.raw*. Sen alla on vastaanotettu tiedosto. Punaiset neliöt osoittavat esimerkin toisiaan vastaavista purskeista. Vastaanotettu viesti voidaan purkaa tarkastelemalla purskeiden pituutta.



Kuva 3: Kuvakaappaus Audacity-ohjelmasta esittää lähetetyn signaalin (ylempi) ja vastaanotetun signaalin (alempi). Punainen neliö sisältää välilyönnin ennen sanaa "is" lauseessa "This is a test."

Kuvassa 4 vasemmalla mitataan ”S”-kirjaimen pituudeksi 54 ms. Oikealla “I”-kirjaimen pituudeksi on saatu 48 ms. Käyttämällä mittaamiseen ohjelmointikieltä kirjainten erottelutarkkuutta voidaan parantaa alle millisekuntiin. Toinen tapa parantaa erottelutarkkuutta on venyttää pulsseja viestiä luotaessa. Esimerkiksi kuvan 2 koodissa *10*ord(t)* korvataan komennolla *100*ord(t).*



Kuva 4: Purskeiden pituus voidaan mitata nauhoitetusta signaalista Audacity-ohjelmalla. "S" -kirjaimen pituudeksi on mitattu 54 ms (vasemmalla) ja "I" -kirjaimelle 48 ms.

Häirintä

Kokelas voi luoda kuuloalueen reunalla olevaa korkea- tai matalataajuuista ininää muuttamalla kohinan halutulle taajuudelle esimerkiksi Pythonin *sin*- tai *cos*-funktion avulla. Näin hän voi häiritä naapureita ja signaalin paikantaminen on vaikeaa.

Estäminen

Tässä dokumentissa esitetyn kommunikoinnin voi estää poistamalla kaikki käyttäjän oikeudet */dev*- ja */proc*-kansioista. Toisin sanoen, käyttäjällä ei saa olla oikeuksia lisälaitteisiin tai sarjaportteihin. Tämä estää myös vaihtoehtoiset vilkkuva ruutu–videokamera, ir-lähetin-vastaanotin, sarjaportti, yms viritykset. Lisäksi terminaalin (myös alt+ctrl+f1), ohjelmointikielien (Python, shell script, awk, yms.) ja tekstieditorien (vi, nano, gedit, …) poistamien tekee koodaamisesta erittäin vaikeaa.

Häirinnän estäminen on helppoa: poistetaan kaikki audiokomennot ja varmuuden vuoksi myös kaikki ruudun valoisuutta säätelevät komennot.

Yhteenveto

Tässä dokumentissa on esitetty periaate, jolla kokelaat voivat keskustella keskenään ja häiritä toisia kokelaita. Esitettyä kommunikointitapaa voidaan muokata hienostuneemmaksi esimerkiksi seuraavin muutoksin:

1. Kantoaalto kuuloalueen ulkopuolelle
2. Napinpainalluksen nauhoittaminen kantoaalloksi
3. Automaattinen salauksen purku (subprocess(arecord) -> algoritmi -> kirjain) joka pyörii koko ajan taustaprosessina

Nämä muutokset pienentävät kiinnijäämisen riskiä. Internetistä voi ostaa 10 eurolla radiotaajuudella toimivia lähetin-vastaanotinpareja. Jos omat läppärit ovat sallittuja, kokelas voi helposti kolvata laitteen pc-speakeriin ja mikrofoniin. Kommunikointi on aina periaattessa mahdollista, jos ohjelmallinen pääsy I/O-laitteisiin sallitaan.