Den Store AV-Bog

Søren Pilgård - DIKUrevyen

21. november 2016

Indhold (kort)

Indhold

1 Introduktion

"Er der en AV-mand tilstede?!?"

Sådan kunne spørgsmålet lyde. Når du har gjort dig bekendt med denne bog er du forhåbentligt istand til at svare *"Ja!"*. Denne bog henvender sig til dig der aldrig har lavet AV i en revy før og skal have en udførlig introduktion, til dig der har lavet AV før men lige skal have lidt hjælp til hvordan det nu var med det hersens AV-system, og til dig der har lavet AV i mange år men stadig har behov for at slå op i dokumentationen i ny og næ.

Dette er en introduktion til DIKUrevyens **AV-system**. Systemet er udviklet af Søren Pilgård, og selv om det startede hos DIKUrevyen har det allerede haft bred udbredelse blandt andet i DIKUrevyen, SaTyRrevyen, Biorevyen og Matematikrevyen. Systemet er af den slags der nok aldrig bliver helt færdigt, der er altid lige noget mere der kan tilføjes, laves om og forbedres, det samme gælder denne bog. Det bør dog ikke afskrække dig fra at kaste dig ud i det.

Dette kapitel samt kapitel 3 forklarer lidt generelt om hvad det vil sige at lave revy og hvad AV-manden laver, samt nogle guidelines til hvordan man gør det godt som AV-mand.

At vi kalder det AV-*mand* skal du ikke tage så nøje, der findes super seje og kreative AV-folk af alle køn!

1.1 Hvordan foregår en revy?

Dette er måske din første revy, eller måske bare din første revy på Natfak. Her vil vi for god ordens skyld kort gennemgå hvad en revy i denne sammenhæng indebærer. De fleste revyer har en nogenlunde ens opbygning dette gælder især for revyerne i SaTyRfællesskabet som indefatter DIKURevyen, Fysikrevyen, Matematikrevyen, Biorevyen, MBKrevyen, og så selvfølgeligt SaTyRrevyen selv. En revy er overordnet delt op i forskellige numre, f.eks. sange eller sketches, disse numrer er, som regel, men bestemt ikke altid uafhængige af hinanden uden overlap i hverken karakterer, tema eller indhold. Der sker dog ofte undtagelser iform af f.eks. følgetoner med fortløbene handling, eller f.eks. sketches der naturligt glider over i en sang. Numrene er grupperet i akter af en halv time til tre kvarters varighed med en 15-20 minutters pause. Der er typisk 2 eller 3 akter efterfulgt af et par ekstranumre.

En klassik revy foregår således at klokken 19.00 åbner dørene til foyeren, typisk er de allerede åbne, men det er her folk begynder at stille sig i kø uden foran dørene til auditoriet. Kl. 19.15 åbner dørene ind til auditoriet og publikum strømmer ind. Når folk har fundet deres pladser sker det ofte at de forskellige studieretninger begynder at synge af hinanden. Dette er en god gammel tradition hvor nye svarvers dukker op og stemningen bliver sat. Det er dårlig stil når revyen går i gang mens folk synger, så man skal sørge for at time det før eller efter en sang. Når klokken er 19.30 og backstage er klar signalerer en boss eller instruktør til TEXnikken at nu kan vi starte. Når der ikke synges sætter lysmanden så revyen i gang ved at dæmpe lyset i salen. Et øredøvende sus vil da gå igennem auditoriet som hele det feststemte publikum fylder med råb, banken og hujen.

Nu er revyen i gang!

De fleste revyer starter med en bandintro, en video der præsenterer bandet, når videoen er færdig sætter bandet sig til rette på bandscenen mens folk brøler *"Bandet!, Bandet!..."* nogle revyer har så en velkomst sketch hvor man præsenterer revyen på en sjov måde (fysik starter f.eks. altid med at aflyse revyen), andre går bare direkte over i en startsang der skal få gang i salen.

Nogle revyer foretrækker at holde 20 minutters pause, men skrive 15

Herefter følger en masse blandede sange og sketches i form af akt 1. Man slutter typisk en akt med en sang med gang i så folk går feststemte til pause. Det er lysmandens opgave at åbne op for lyset, men AV manden starter typisk med at have et pauseskilt/billede/video til at fortælle at nu er der pause først. Herefter er der pause i 15-20 minutter mens publikum køber øl og går på toilettet. (Som AV-mand kan det også være en god ide at benytte toiletterne nede backstage og fylde op på vand/drikkevarer nu). Lidt før pausen er klar begynder bandet at spille ind, de spiller typisk noget musik der passer til deres tema eller som de har brugt mens de øvede sig.

Så starter næste akt (hver opmærksom på at der altid er et par stykker der kommer forsent og som konsekvent glemmer at lukke dørene (så hav en løber klar)). I en 3 akters forestilling kører andet akt stille og roligt (sådan konstruktionsmæsigt), i en 2 akters svarer det til at gå direkte til 3. Hver opmærksom på at nogle 2 akters forestillinger kan have en tendens til at have lidt længere akter. Sidste akt slutter af med et slutnummer, en god sang der får folk op og køre. 3/4 inde i sangen kommer alle skuespillerne ud og der siges tak til alle der skal have tak og alle på scenen bukker et par gange. Så lunter folk ud af scenen og lyset går ned så der bliver helt mørkt. Imens klapper folk ihærdigt og råber *"Ekstra nummer!, Ekstra nummer!..."* for det siger traditionen. Det er ofte enten AV eller lys der starter ekstra numrene man skal altid trække den lidt, så trække den lidt mere, og så liiiige trække den lidt mer', både fordi sådan er det, samt for at de kan nå at skifte kostumer/gøre rekvisiter klar. Og så går ekstranumrene i gang. Disse består som regel af 3-4 numre, nogle gange en video og slutter af med en slut-slutsang. Denne sidste sang er handler ofte om at nu er revyen slut, men der er efterfest bagefter og så skal den have fuld gas. Så er revyen færdig, man kan smide et skilt op om at der er efterfest. Hvis man har flere forestillinger men kun én efterfest kan man smide en henvisning om at gå på *Caféen?!* op.

Generelt prøver man ofte at holde det intelligente humor i første akt hvor folk stadig kan forstå det og ikke ville "kede" sig, mens 3 akt er mere plat humor, dårlige ordspil og billige punchlines. I takt med at publikum bliver mere og mere fulde passer dette også meget fint til hvad de magter. Publikum kan være helt fantastiske, men de kan også være larmende, forstyrende og ret overvældende. Dette gælder både for dem på scenen, men det kan også være skræmmende når TEXnikken laver en fejl og hele auditoriet brøler *"TeXnikken sejler"*. Det vigtigste er at holde hovedet koldt i en hver situation. Få ro på, tænk igennem hvad der går galt, lad være med at opildne publikum, fokuser, og få tingene tilbage på sporet. Vi har **ALLE** været derude hvor det hele gik galt, og bagefter gik det hele altsammen, bare husk på ovenstående.

Hvis du er i tvivl om et begræb så tag et kig i kapitel 1.3.

1.2 Hvad laver en AV-mand?

Hvad er det så AV-manden laver til en revy? Jo, ser du, TeXnikken består af 3 dele, der er lyd, de sørger for at bandets musik, sangernes sang og skuespillernes grynten kommer ud af højtalerne. Lydmanden laver så at sige ikke noget lyd selv, han sørger

bare for at alle kan høre det og at det lyder godt. Lysmanden sørger for at man kan se noget, det gør han med et væld af farvede lamper og spots, med lys og skygge samt en gang taktisk røg kan han transformere stemningen og skabe en fest på scenen.

Nogle revyer er også begyndt at have en Epelipsiadvarsel, i starten af revyen og før blinkende numre

AV-mandens rolle er mere alsidig, en slags digital rekvisit man kan bruge til lidt af hvert. Det kunne f.eks. være at agere powerpoint-show til en "fremvisning", vise billeder til at illustrere forskellige pointer, vise videoer, vise pauseskilte, vise hvilke sponsorer der er, afspille digital musik til dansenumre, afspille lydeffekter som pistolskud og prutter. En af de vigtigste opgaver er dog at lave overtekster. Det kan være utroligt svært at lave god lyd i Store UP1, hvis sangerne samtidigt er lidt usikre og der er en masse larm i auditoriet kan det være utroligt svært at høre hvad der bliver sunget. Samtidigt er der mange sange hvor man gerne vil give publikum mulighed for at synge med i omkvædende, derfor har man valgt at indføre overtekster. Overtekster er ligesom undertekster på en film, de sungne tekster bliver bare vist på et hvidt lærred oppe bag scenen for at alle kan se dem. Dette har vist sig at være en stor success.

Pas på dig selv! Det er nemt at brænde sig på et stort projekt.

Hvis man er kreativ og har teknisk kunnen kan man også bevæge sig ud i mere avancerede ting, animationer af alle mulige slags som passer til hvad der sker på scenen. Revy-systemet understøtter en masse seje ting, og der kommer hele tiden flere til så hvis man har mod på lidt programmering er det bare at give sig i kast med effekterne. Men pas på, lad være med at bide over mere end hvad du kan sluge. Alle disse avancerede detaljer er lækkerier der kan oppe oplevelsen, men de kan aldrig stå alene og hvis du ser det som en for stor opgave skal du ikke være bange for at spørge om hjælp eller sige fra.

Men når det er sagt: Man kan kun skabe noget nyt ved at prøve nye ting af.

1.3 Hvad er hvad?

Her følger en række gængse termer der kan være praktiske at bruge, og andre småting der er værd at huske.

Scene: Til hver revy sættes en stor hjemmebygget træscene op i bunden af Store UP1.

Bandscene: En lille forhøjet scene til venstre for scenen, hvor bandet sidder og spiller.

Bagtæppet: Et tæppe til at aflukke til den bagerste meter af scenen, så man kan stille sig klar uset.

Højre: Højre side af scenen set fra TeXnikken og publikum. Den side hvor folk oftest kommer ind fra. (Skuespillere kan ikke finde ud af højre og venstre)

Venstre: Venstre side af scenen set fra TeXnikken og publikum. Den side hvor bandet sidder (Skuespillere kan ikke finde ud af højre og venstre)

Trekanten: Området til højre på scenen bag tæpperne. Hvor skuespillerne kan stå klar og let komme ind fra siden.

TeXnikken: Lyd, Lys og AV, samt diverse hjælpere. Kortsagt dem der sidder foran scenen og arbjeder under revyen

Backstage: Området bag scenen. Bruges også om dem der arbjeder der. De andre "skjulte" roller som dem der laver rekvisiter og kostumer.

Rekvisitten: Dem der laver rekvisiterne/stedet de laver dem (backstage)

Rekvisitkælderen: Revyernes fælles opbevaringslokale, backstage under bandscenen. (Pas på skimmelsvamp)

Dyrestalden: En del af DIKU, gangen der ligger lige når man kommer ud af Store UP1 fra scenen.

Omklædningen: Backstage bag de store metaldøre og indtil trædørene står skuespillernes ting under revyen, gør det svært at komme igennem!

Hyggeområdet: Området bag skuespillernes omklædning, for foden af den sydlige trappe. Her findes ofte snacks og drikkevarer under revyen (loot det i pauserne)

Harlem: Et "hemmeligt" lokale i DIKUs kælder som revyerne råder over. Her øver bandet indtil de rykker ind i auditoriet. Her står også det meste af den texnik som revyerne råder over samt kasser med kabler. Sørg for at spørge før du låner størrere ting, husk at ryde op efter dig, husk at tape kabler sammen inden de dumpes. Spørg din revyboss om adgang hertil

Indre Harlem: Inde i Harlem findes indre Harlem der bruges til at redigere videoer samt de optagne film fra forestillingen. Her står det udstyr vi passer ekstra godt på. Kun de særligt betroede får adgang hertil og man skal spise en skovsnegl og kysse en datalog for at få lov!

Adgang: Husk at snakke med din boss om at få aktiveret dit studiekort så du kan komme ind på DIKU og i Harlem. I TeXnikken kommer og går man ofte på skæve tidspunkter så det er praktisk med adgang.

Instruktør: Dette er meget forskelligt fra revy til revy, men typisk en kreativ/kuntnerisk/skuespilsmæsig ansvarlig for de forskellige numre

Boss: Ansvarlig for hele revyen (bestemmer)

Universitetsparken 1: Stedet der i mange år har huset DIKU (i hvertfald de studerende). Skal forlades i år ....

Store UP1: Det store auditorium på Universitetsparken 1, også kendt som store knirke. Det er her alle revyerne på natfak afholder deres revyer.

Caféen?!: En af fredagsbarene på Natfak. Mange efterfester holdes her, og ofte sender man publikum til Caféen?! om fredagen efter første forestilling/generalprøven.

Overtex: Et stort hvidt område over scenen under loftet hvor AV projekterer det meste af sine ting

Højtex: Et fancy begræb for alle de projektioner der er over Overtex på selve loftet

Lavtex: Et fancy begræb for alle de projektioner der er under Overtex, altså ned på selve scenen (antagelivis på et lærrede der opstilles undervejs)

Fisk: Et fancy ord for en kort film der ligger imellem to numre. Bruges ofte som et sjovt indslag nogen kom på samt til at give lidt ekstra tid til at nå et svært kostume/sceneskift.

Datamat: Fysisk implementation af Turings abstrakte maskine, hvad man på Engelsk ville kalde en "computer".

LaTeX: Et "sprog" til at skrive og opsætte tekst, tænk HTML men til at lave nydelige artikler, formler og pdf'er. Også brugt til at lave overtekster.

TeX: Forgængeren som LaTeXer bygget på, de to termer bruges oftest synonymt.

Emacs: Gammelt tekstredigeringsprogram (Tænk notepad, men mere avanceret end Photoshop). Kan udvides helt utroligt og danner basis for DIKUrevyens AV-system.

Script: Et kort program, en sekvens af instruktioner for hvad der skal ske (automatisk) når det bliver kørt.

Lisp Et gammelt programmeringssprog. Der findes mange varianter men det er kernen i Emacs og udgør dermed måden man laver scripts på i AV-systemet

1.4 Forudsætninger

Note: TODO: Dårligt afsnit. Find ud af hvor det passer, og omforumler det til det her skal du lære

Hvad skal man kunne for at bruge DIKUrevyens AV-system?

- LaTeX
- Linux
- Emacs
- Emacs Lisp
- Python

LaTeX
Bruges til at skrive selve overteksterne. Der bruges Beamer til at lave et langt slideshow, overtex.sty definerer en række makroer der gør det let at lave en lang præsentation. De fleste på natfak burde kunne LaTeX, hvis ikke er den mængde der bruges til at lave normale overtekster ret lille og burde kunne mestres ved bare at kigge i nogle af de gamle filer.

Emacs
Emacs bliver brugt som kontrolcenter til det hele. Det er derfor væsentligt at have en hvis forståelse for hvordan Emacs virker. Emacs er et voldsomt konfigurerbart program til at arbejde med tekst. Jeg er selv stor Emacsbruger og har opbygget et helt unikt system. På sigt er det håbet at lave en standard konfiguration som folk der ikke er emacsbrugere kan udnytte. En sådan konfiguration ville kunne skjule at det overhovedet er emacs der kører bag det hele.

Emacs Lisp
Emacs Lisp er det primære scripting sprog der bliver brugt. Det bruges til at automatisere ting i Emacs, derudover kan der kaldes externe kommandoer og kommunikere

med andre maskiner. Selve revysystemet bruger en stor del Emacs Lisp så hvis noget stopper med at virke er det godt at kende. Til almindeligt AV brug kan man dog nøjes med en stærkt begrænset del som denne guide nok skal introducere.

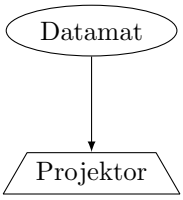
Linux
DIKUrevyens AV-system er bygget og kører på linux styresystemet. Det betydder at man for at bruge systemet effektivt er nød til at have en hvis forståelse for at bruge en linuxmaskine igennem en terminal. Et håb er en gang at have en form for standard opsætning af datamater hvor alt nødvendigt er installeret, som AVmand skal man så blot sætte udstyrret op og lave indholdet.

Python
Bliver brugt til de forskellige programmer og værktøjer der udgør resten af AV-systemet. Det burde kun være nødvendigt at kunne kode python for at udvikle/-vedligeholde systemet.

2 Tanken bag

Den grundliggende filosofi bag systemet er at AV skal kunne komplementere eller understøtte en revy uden at komme i vejen. Der skal dermed kun vises det som publikum skal se og høre og intet andet. Det kan være med til at bryde indlevelsen og virke utroligt amatøragtigt lige så snart publikum ser en cursor, en film der maksimeres eller rammerne på et vindue. Det er sådanne fejltagelser der hiver folk tilbage til virkeligheden i auditoriet fremfor den fiktion man skaber. Man skal ikke tænke over at det er AV, men at det er en mega fed revy, folk bør således kun lægge mærke til AV når der sker noget ekstraordinært, og forhåbentligt ikke på grund af noget der ikke skulle ske.

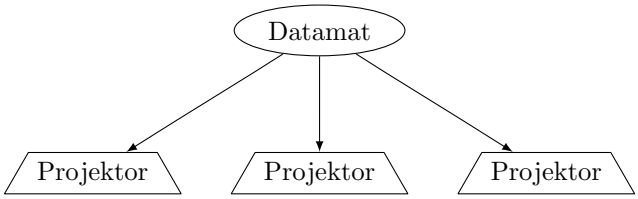
Det simpleste AV-system man kan lave (og som er det de fleste bare ville gøre uden at tænke videre) er at have en datamat tilkoblet en projektor. Datamaten kan derfra køre et presentationsprogram, f.eks. powerpoint.



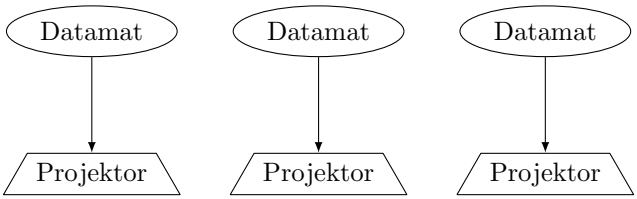
Figur 1: Simpel AV opsætning.

Dette er dog ret primitivt. Det er utroligt nemt at komme til at lave fejl, så som lige at få skubbet cursoren hen på et andet desktop eller at filmen spiller på den forkerte skærm. Ting der er sket gang på gang til mange revyer, og som hurtigt får publikum til at råbe *"TeXniken sejler!"*.

Derudover har man et problem hvis man skal bruge mere end 1 projektor.



Figur 2: En datamat, flere projektorer.

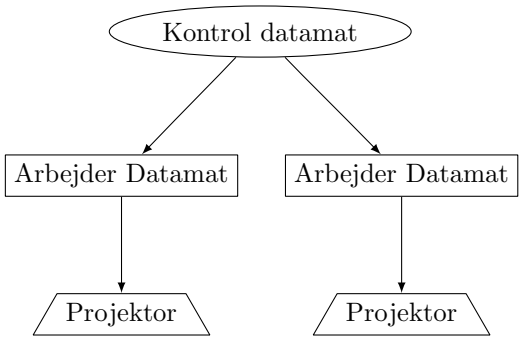


Figur 3: Flere datamater, flere projektorer.

Man kan enten koble flere projektorer på en datamat, hvilket kræver en datamat der er i stand til dette, hvilket ikke særligt mange er. Og det kan give en hovedepine hvis

man samtidigt prøver at holde det "skjult". Alternativt kan man have flere datamater koblet til hver sin projektor. Nu skal man så bare navigere rundt mellem en helt masse maskiner eller have en AV-mand pr. maskine hvilket heller ikke er særligt praktisk. Desuden har vi stadig problemet med at man let kommer til at dumme sig på samme måde som i den første opsætning.

Det vi i virkeligheden ønsker os er en abstraktion mellem det at *styrre* AV og det at *vise* AV. Hvis tingene kører adskilt og har klart definerede ansvarsområder kommer man ikke lige så let til at lave fejl. Vi ønsker derfor at have en central maskine der styrer det hele, denne kommunikerer med andre maskiner der sørger for at vise ting via projektorer.



Figur 4: Central styring, flere arbejdere med hver deres projektor.

Det er dette princip DIKUrevyens AV-system benytter sig af.

Note: Lidt baggrundshistorie for de nørdede.
Det første hjemmelavede AV-system blev lavet af Troels Henriksen. Han lavede et programmeringssprog *Sindre* i Haskell til at lave grænseflader. I det sprog lavede han et `dmenu` lignende program kaldet *sinmenu* som han brugte som interface. Selve overteksterne bestod af en pdf som han lokalt oversatte til rå tekst som blev fodret ind i et script der brugte hans grænseflade. Dette script kommunikerede med en server koblet til en projektor over en sshforbindelse. `Xpdf` kan køres som en server der kan modtage kommandoer fra en kommandolinje, på denne måde kunne man derfor styre overtekster over ssh.

Desværre skalerede løsningen ikke særligt godt. Da der begyndte at komme mange AV-effekter kunne man ikke både køre overtekster og andet AV alene.

Dette blev (i hvertfald forsøgt) rettet op på da jeg (Søren Pilgård) efterfølgende begyndte at skrive et nyt AV-system til DIKUrevyen 2012. At skrive et AV-system er dog ikke en nem tjans og det endte med at foregå i en løbende process med forbedringer til hver revy.

Note: Herfra skal der skrives om, det er mere en "se hvad jeg har lavet" end et "nødvendigt mindset"

DIKUrevyens AV-system består af en central grænseflade der kan kommunikere med flere forskellige maskiner der kan vise AV materiale.

Selve grænsefladen er udviklet som en række udvidelser til Emacs. Grunden til dette er at da jeg startede indså jeg at systemet skulle kunne følgende ting:

- Kommunikere med en server der viser indholdet.
- Det skulle kunne vise ting overskueligt i en grafisk grænseflade.
- Der skulle være mulighed for at indlejre scripts så man kan køre ting automatisk på bestemte steder i forestillingen.
- Det ville være praktisk hvis man kunne rette fejl i texkoden mens man viste pdf'en da man ellers risikere at glemme dem.

Det gik hurtigt op for mig at det ville være fjollet at udvikle noget fra bunden da Emacs i forvejen kunne meget af dette. Desuden er jeg Emacsmand og så hurtigt hvordan en integration ville være nice. Den grundliggende formel for at vise overtekster er at man åbner et LaTeX dokument der bruger beamer pakken til at lave overteksterne. Så starter man `ubertex`minor modet, dette sørger for at pdfen bliver lagt op på serveren. Herefter skjules de fleste LaTeXkommandoer og et overlay lægges der viser hvad der bliver vist. Man kan så trykke "næste" hvilket rykker overlayet ned og synkroniserer serveren til at vise det tilsvarende slide i pdfen. Man kan også trykke vilkårlige steder i tex filen og rykke direkter hertil i overteksterne. Desuden kan der indsættes kode der eksekveres når man når til det pågældende slide.

Derudover findes minormodet `uberscript` der lader en afvikle scripts. Det kunne f.eks. være en sketch med en række lydeffekter eller et kald til en video. Et script kunne også være aktoversigten hvorfra man ved at trykke "næste" kommer ind i det næste nummer, og når dette er færdigt kommer man så tilbage og er klar til næste.

Hvis alt går som det skal, skal man som AVmand kun trykke på en knap (næste) for at afvikle en revy. Dette må være essensen af et godt AV system, når man kun skal tage sig af timingen på skuespillerne/sangerne samt disses fejltagelser.

3 - Guidelines til en AV mand

TODO: Her kommer der til at være en række mere generele råd om hvordan man laver god AV

Det vigtigste man skal huske for at være en god AV-mand, er at have det sjovt! Hvis du ikke har det sjovt, bliver det ikke en sjov revy. Så lad være med at stresse for meget, ignorerer når de andre sejler og glemmer at give dig hvad du skal bruge og hyg dig. Hvis du først lader dig blive presset af det hele mister du overblikket og kreativiteten, så hellere sige "Det går nok" og tage det som det kommer, det skal nok blive en revy.

I dette kapitel findes en række råd og guidelines. Hvordan man laver AV er en smagssag og afhænger meget af hvad man laver og sammenhængen. AV er således en kunst, og selvom det der står her kan lyde som regler, ved enhver kunstner hvornår man skal bryde dem. Nogle gange bryder man reglerne for at få tingene til at gå op i en højere enhed, andre gange for at lave et helt nyt regelsæt. Selvom du måske er anarkist og vil gå din egne veje som AV-mand vil jeg dog anbefale dig at læse og forstå hvad der står her.

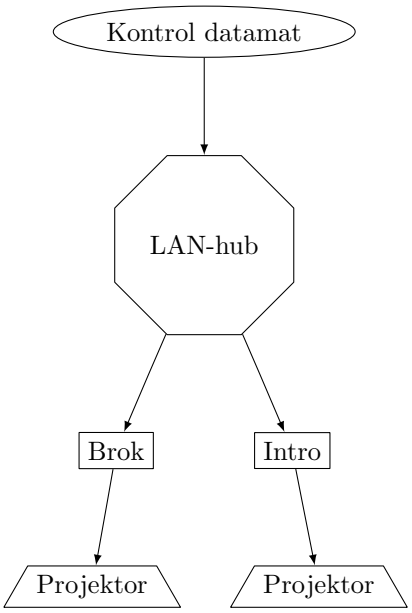
Note: Husk der er mange måder at lave AV på, det her er nogle af de erfaringer der har virket godt

Note: Sæt altid højere standarder til dig selv end andre

3.1 Om at lave gode overtekster

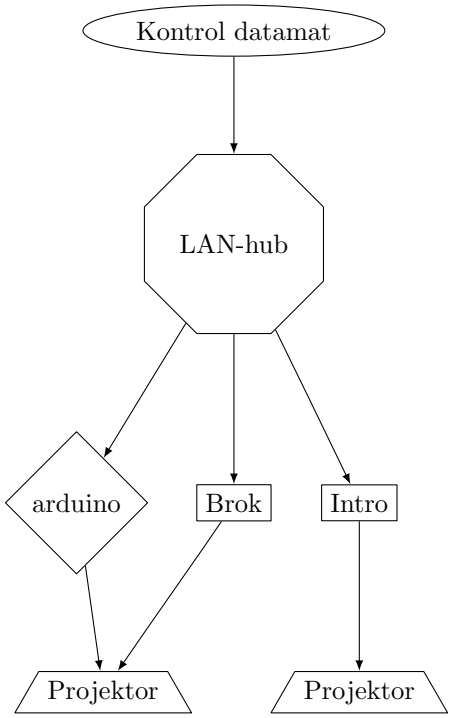
4 - Opsætning af tekniken

Til en standard opsætning ala DIKUrevyens skal du bruge:



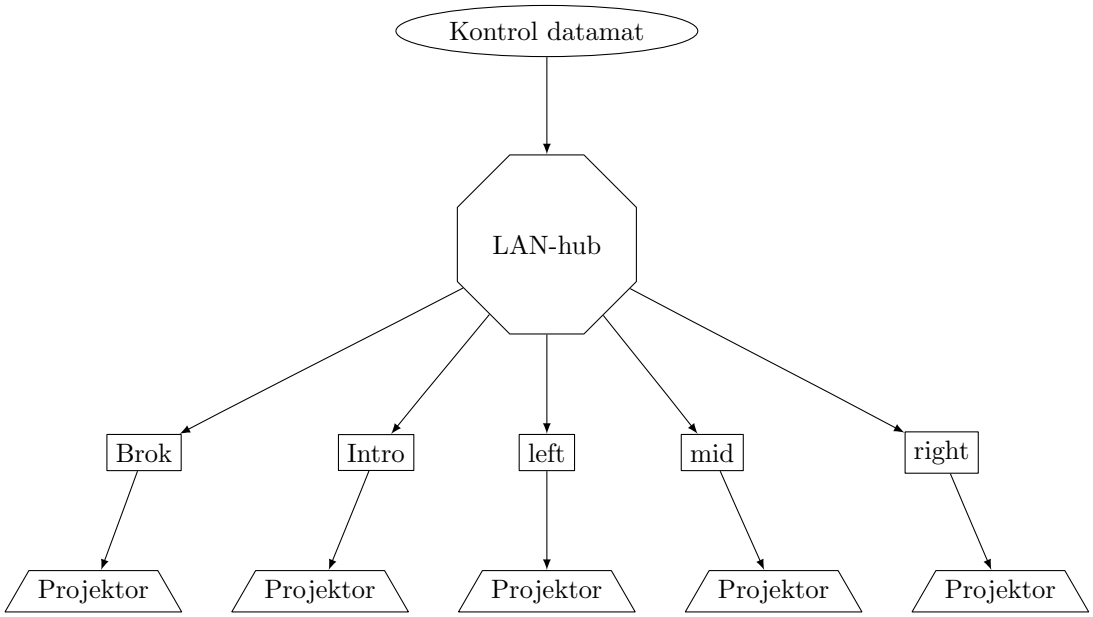
Figur 5: Standard opsætning

Hvis det ønskes (Når det er færdigt) kan man køre med fjernstyrede projektorklapper.



Figur 6: Standard opsætning, med arduino til projektorklap

Systemet kan udvides, her ses f.eks. en opsætning med højtex (uden projektorklapper) som brugt til DIKU Jubilæumsrevy.



Figur 7: Højtex

4.1 Kontrol

Dette er din primære indgang til systemet. Jeg anbefaler at man bruger en Foldedatamat, gerne ens egen bærbare. En bærbar har den fordel at skærmen kan indstilles til både at sidde ned og stå op. Derudover har den tastatur og mus indbygget så det ikke fylder i den ellers rodede texnik. Og så kan man tage den med sig så man kan arbejde videre andre steder. F.eks. er det praktisk at kunne plugge den ud så man kan arbejde videre på sine overtekster til et ellers kedeligt senemøde i hyggehjørnet.

4.2 Projektorer

Til DIKUrevyen bruges der typisk i hvertfald 2 projektorer. Der bruges én der peger op på lærredet over scenen, her vises der overtekster, almindelige av-ting til sketches samt små film.

Kantinen ejer en stor Epson som bruges til introen/av på scenen

DIKUs projektorer bruges til alt andet. DIKU har en række forskellige projektorer, hvor det kan være svært at kende forskel på en del af dem.

En god tradition, som en AV-mand bør holde i hævd, er at rense projektornes filtre når man henter dem i begyndelsen af revyugen. Der er tilsyneladende ikke andre der gør det, så lad det blive en del af rutinen. Så overopheder de ikke lige så nemt.

Til projektorene er der bygget en række projektorkasser af gamle colakasser. Disse gør det en del nemmere at indstille projektorene.
I gamle dage, da jeg var ond. Da blev projektorene stablet på bøger til de stod sirligt, den ærede AV-mand Troels Henriksen brugte mangt en stund på at bande og svovle når disse blev rykket
Nu gør projektorkasserne det en del nemmere da man kan stripse/tape kasserne fast og det hele bliver langt mere stabilt. Desuden kan der komme langt mere luft til.
I gamle dage, da jeg var ond. Da blev projektorne så varme at de overophedede, så vi måtte til HCØ og hente tøris til køling. Det gav også kolde drikkevare (til tider frosne).

Se afsnittene om Ubertex, Uberscript og Emacs for at finde ud af hvordan softwaren bruges.

4.2.1 Projektorklapper

En ulempe ved projektore er at deres "sorte" ikke er mangel på lys, men bare *mindre* lys. Det betyder at når alt lyset i StUP1 er slukket og en projektor står og lyser, er der stadig ret meget lys på scenen. Det ser **MEGET** dumt ud og gør at man kan se hvad der sker på scenen. Det er primært et problem for projektoren til overteksterne og kan accepteres til f.eks. højtex (da loftet ikke er lige så reflekterende)

Det er desværre ikke en løsning at slukke/tænde projektorne da det tager for lang tid/er besværligt/ skydder farver op når de tændes.

Derfor bruges en 'projektorklap'. Der har i mange år været brugt en halv papkasse på en stang. Det fungerer, men det kan godt være lidt stressene. Man skal huske at få klappen på når man er færdig med en sang og har travlt med at huske hvad der nu

skal ske. Til tider sker det også at man glemmer at tage klappen af, man når typisk at panikke lidt når der ikke kommer noget billede frem. Det kan betyde at man misser de første par overtekster eller starten af en film.

For at løse dette er det planen at der bygges nogle arduinoer der kan kobles på netværket, disse kontrolerer en klap foran projektoren. På denne måde kan projektoren automatisk begynde at skyde billedet op på lærredet Arduino ftw!

4.3 Netværk

4.3.1 Statisk IP

Hvis dit netværks interface hedder enp0s25

```
sudo ip link set enp0s25 up
sudo ip addr add 192.168.0.XXX/24 dev enp0s25
```

alternativt

```
sudo ifconfig eth0 192.168.0.XXX
```

4.3.2 Hosts

Da det kan være svært at huske alle ip-addreserne kan man i stedet navngive maskinerne. Dette gøres lokalt i `/etc/hosts` et eksempel:

```
192.168.0.20    intro
192.168.0.30    brok
192.168.0.40    left
192.168.0.50    mid
192.168.0.60    right
```

Nu kan man ssh'e ind ved blot at skrive:

```
ssh brok
```

Det virker også med scp ol.

4.3.3 Forbindelse uden login

For at logge ind på systemerne over netværket bruges ssh. Da det bliver jævnt irriterende hele tiden at skulle taste løsener kan man lægge sin offentlige ssh nøgle ind på de forskellige maskiner

```
cat .ssh/authorized_keys | ssh revy@192.168.0.30 "cat >> ~/.ssh/authorized_keys"
```

det kan være du først skal oprette mappen `.ssh`.

alternativt kan du bruge

```
ssh-copy-id revy@192.168.0.30
```

Udskift `pilen` og ip'en med den relevante bruger og ip. Husk at du skal have en ssh nøgle først dannes med:

```
ssh-keygen
```

4.3.4 ssh mount

I stedet for at scp'e alt muligt crap frem og tibage kan man benytte sig af et ssh mount, som er et filsystem over ssh. Jeg anbefaler at man har filerne liggende lokalt og fra hver maskine der skal kommunikeres med køres `sshfs` som får filerne frem.

Installer sshfs F.eks.:

```
sudo pacman -S sshfs}
```

fuse `sshfs` benytter fuse

```
sudo modprobe fuse
```

4.3.5 Gateway

Hvis du er intereseret i at få internet på det lokale netværk kan man lave en gateway. Dette består i at en af datamaterne på det lokale netværk også er forbundet til et netværk med internet, f.eks. eduroam.

Der er en guide på `sigkill.dk/writings/guides/gateway.html`

På gateway datamaten sættes først den statiske ip (allerede gjordt i forrige trin) Her efter køres gateway.sh som root (hentes på siden)

På clienterne der vil udnytte gatewayen sørger man først for at der er en statisk ip.

herefter skrives

```
route add default gw 192.168.0.XXX
```

eller

```
ip route add default via 192.168.0.XXX
```

Hvor "192.168.0.XXX" er ip'en til gatewaymaskinen

herefter sættes en dns server, f.eks. google

```
echo nameserver 8.8.8.8 > /etc/resolv.conf
```

Hvis dette ikke virker kan man evt. kigge på `https://wiki.archlinux.org/index.php/Internet_Share` for en ip baseret løsning

4.4 Arbejdere

Arbejdere er de maskiner/servere der er sluttet til projektorne. Det er deres rolle at vise AV-indholdet det kunne være overtekster, film, lydeffekter, slides, billeder osv.

På nuværende tidspunkt bruges `mplayer` til at vise film, `feh` til at vise billeder og `xpdf` til at vise pdf'er. Det smarte ved `xpdf` er at det kan startes som en server som kan modtage kommandoer via en terminal.

Ideen er at man fra sin kontroldatamat kan ssh'e ind på arbejderne og afvikle den kommando man skal bruge. Emacs med ubertex/uberscript tilbyder basalt set et interface til at gøre dette nemt.

Det er planen at programmet `Zeigen` udvikles. Dette skal erstatte `feh` og `xpdf`. I stedet for at skulle ssh'e ind lytter `Zeigen` til en port og udfører kommandoer på givne klokkeslet. Så kan man sige "om 1/2 sekund skal du afspille denne video" på 3 forskellige maskiner, hvorved videoen vises simultant via `mplayer`.

Et eksempel på en arbejderdatamat er `brok` der bruges til overtekster og ting der skal vises på lærredet. Derudover findes `left`, `mid` og `right` der bruges til at vise *højtex*.

4.5 Brok

Her er en guide om at få brok til at virke:

Sæt ham til Projektor, tastatur, netværk og strøm.

Hvis der skal spilles lyd fra brok bør der være jord på da der ellers kan komme en brummen.

Brok har på skrivende tidspunkt ubunut 11.04 natty. Det betyder også at der er gnome, unity og ting og ækle sager på. Disse har en tendens til at gøre livet surt, da de overskriver xorg konfigurationer og gør mærkelige ting man ikke helt kan gennemskue.

Dette er IKKE optimalt! Support til 11.04 udløb oktober 2012. Der bør installeres et ordentligt styresystem. UDEN gui (installeres seperat).

1: Tænd brok
Brok logger automatisk ind med brugeren ***revy*** med løsnet ***hamster*** Herefter starter unity der har en 'kør' dialog ting.

2: Start en terminal
Skriv `terminal` og tryk enter. Der vil nu komme en terminal op i hjørnet.

3: Kom på netværket

```
sudo ifconfig eth0 192.168.0.30
```

Giv ham en passende ip du kan huske.

4: Opret SSH forbindelse Der kører allerede en ssh daemon. forbind til ***revy*** med løsnet ***hamster***. Du kan evt. uploade din offentlige ssh-nøgle så du slipper for at logge ind.

```
ssh revy@192.168.0.30
```

5: Stop gdm
Gnome Desktop Manager skal lukkes.

```
sudo service gdm stop
```

Dette lukker hele den grafiske grænseflade, inklusiv X.

6: Start screen

```
screen
```

screen køres så vi kan starte X i baggrunden. Når programmet startes vises der en menutekst, bare tryk enter. `screen` kører nu.

7: startx
Start X manuelt.

```
startx
```

Dette starter X op hvorefter .xinitrc eksekveres.

Som det er lige nu startes `xmonad` som windowmanager. Det betyder at skærmen pr. default er sort når x er startet

8: detach fra screen
Tryk `Ctrl-a d`. Dette lader alt der kører i screen fortsætte upåvirket mens du kommer tilbage til det forrige miljø. for at reattache skriv da `screen -r`.

9: test xmonad
Test om xmonad virker. På broks tastatur trykkes `Alt-Shift-Enter`. En terminal burde åbne sig. Luk igen med `Alt-Shift-c`.

Læs evt. afsnitet om xmonad.

Da terminalen er en default `xterm` med hvid baggrund er dette et snedigt trick til at se hele området projektoren kan lyse op. Jeg bruger dette ofte når jeg tweaker projektoren.

10: prøv ting af Med en ssh forbindelse åben, lad os prøve om vi kan få noget til at virke. Start med at vælge det 'display' der skal vises grafiske ting på

```
export DISPLAY=:0
```

Dette skal køres for hver gang du ssh'er ind.

start nu xpdf med en af de gamle overtekst filer der ligger et eller andet sted.

Se sektionen om kommandoer for nærmere detaljer

Hvis man gerne vil lave flere ting simultant på brok kan man sagtens åbne flere lokale terminaler og ssh'e ind parallelt.

4.5.1 Ting der køres på brok

For at hacke uden om alt muligt gøjl køres et par scripts gennem `.xinitrc`

swarp
`swarp` er et program der flytter musse-markøren (cursoren) til et givent koordinat. Swarp findes på `tools.suckless.org/swarp`. `swarp` findes desuden i arch's repository, det kan være den også findes til debian baserede styresystemer, potientielt i en suckless pakke.

`swarp` køres på brok med argumenterne 20000 20000, hvilket flytter markøren ad helvede til.

istedet for `swarp` kan man bruge `unclutter` med kommandoen `unclutter -idle 0 -root &` Som i fjern cursoren efter 0 sekunders delay efter bevægelse inklusiv når cursoren er over rod baggrunden (altså ikke kun over vinduer).

fixsleep `fixsleep.sh` er et script der forsøger at forhindre X i at slukke skærm outputet. X bruger et system der hedder DPMS (Display Power Management Signaling) til automatisk at slukke skærm outputet efter en periode uden tastaturaktivitet.

fixsleep benytter følgende to kommandoer

```
xset dpms 0 30000 40000
xset s 30000
```

den første linje dækker over `xset dpms [standby [suspend [off]]]` den anden linje dækker over `xset s [timeout [cycle]]`

keepon

```
xset dpms force on
```

Denne kommando forcer dpms fra, (det svarer til at at trykke på en tast)

`keepon.sh` er et script der ligger i baggrunden og kører denne kommando whert 30 sekund.

fixsleep og keepon forsøger begge at holde dpms stangen ved at lave aktivitet. Man kan måske istedet bruge

```
xset -dpms; xset s off
```

Til at slå dpms fra. De to systemer kan dog ikke blandes.

Man kan se om dpms er slået til ved at kalde `xset -q`

4.6 Xmonad

Windos/super eller alt som modifier knap.

5 - Master

5.1 ubersicht

5.2 ubertex

5.3 Konvertering fra manuskrift til overtekster

5.4 Kommandoer

Dette er en liste af kommandoer der kan bruges på masteren. En del af disse kan også bruges interaktivt

5.4.1 Lokale kommandoer

Kommandoer der gør ting lokalt på masteren.

`{revy-start}`
Starter en revy, den første kommando i en .revy fil. Forbinder til alle arbejderne og starter programmet. Hvis den fejler er det typisk fordi en server ikke har forbindelse til netværket (måske er den ikke tændt), fordi masteren ikke har forbindelse til netværket (har du sat stikket i / fået en statisk IP?) eller fordi programmet er installeret i en anden mappe end den angivne.

Note: Der er lige nu en fejl i programmet der gør at denne kommando nogle gange skal køres to gange for at virke

`{revy-quit}`
Lukker programmet på alle arbejderne hårdt og brutalt. Den inverse funktion af revy-start. Bør kun bruges fordi tingene virkeligt er gået i hårdknude, f.eks. hvis en arbejder stopper med at svare når man prøver at eksekvere kode på den.

`{revy-open filename &optional worker}`
Åbner en fil i revysystemet og eksekverer den i dens revysammenhæng. Se de specifikke revy modes5.1, 5.2 for en bedre indføring i de enkelte dele. Når en fil åbnes på denne måde, vises den i en buffer i det korrekte mode og eksekveringen starter fra toppen. Rækkefølgen af filer der åbnes, og deres placering huskes, så et kald til `revy-end-sketch` fra den åbnede fil returnerer til her.

"`filename`" er en tekststreng med navnet på filen. Umiddelbart bruges en relativ sti `(revy-open "sange/lalala.tex")`, og denne sti tager så udgangspunkt i mappen hvor revyen er gemt (på din lokale maskine). Alternativt kan man bruge en absolut sti hvis man f.eks. pludselig fik lyst til at åbne en fil fra en gammel revy.

"`worker`" er navnet på en arbejder, denne vil blive brugt som "default worker" istedet for den nuværende. Så kan man f.eks. åbne en tex/pdf på en anden arbejder end den nuværende `(revy-open "sange/lalala.tex" 'mid)`

"`revy-nop &rest..`" Gør ingenting, og ignorerer alle argumenterne. Kan f.eks. bruges til at udkommentere en kommando. Kan også bruges i .revy filen istedetfor `revy-open` for at signalere at dette nummer ikke indeholder noget AV.

```
(revy-open "sketches/den-sjove.sketch")
(revy-nop "sketches/den-kedelige.sketch")
(revy-open "sange/sjov-sang.tex")
```

`{revy-end-sketch}`
Lukker et nummer. Kan bruges i alle slags filer, både .tex og .sketch. Returnerer til den fil der har åbnet den nuværende og fortsætter fra hvor cursoren stod sidst. Bruges typisk til at komme tilbage til .revy oversigten fra en sang eller sketch.

Funktionen kalder `revy-abort-all`, og lukker dermed allt kørende kode på arbejderne automatisk.

`{revy-return}`
Virker lidt lige som `revy-end-sketch`, returnerer til det sted der åbnede det nuværende nummer. Men lukker ikke kørende kode. Den kører altså ikke `revy-abort-all`.

`{revy-restart}`
Genstarter det nuværende nummer. Svarer til at lukke og åbne filen.

5.4.2 Generelle kommandoer

`{revy-abort-all}`
Den store røde stop knap! Stopper alt der kører på alle arbejderne. Alle lyde, billeder, pdf'er og kode der kører afbrydes. Er praktisk at have i baghånden til når tingene går galt, og kan bruges interaktivt til dette. Men kan også bruges inde i en sketch til at stoppe alt der køres samtidigt. Vær opmærksom på at lyde stoppes ret abrupt, hvilket ikke altid er hvad der ønskes. Bør ikke bruges som panik knap hvis f.eks. en sanger synger forkert, i så fald stoppes hele pdf'en med at blive vist. Så er man nødt til at genåbne filen, finde tilbage igen, og fortsætte. Til dette formål bør man istedet bruge "blank" funktionerne.

`{revy-abort}`
Lige som `revy-abort-all`, men stopper kun de ting der kører på den nuværende arbejder. Så den kan f.eks. bruges i en sketch til at holde højtex kørende mens man stopper et billede og en lyd på overtex.

`{revy-blank)}`
Stopper visningen af grafik på den nuværende arbejder indtil næste input. Alt kode og lyd fortsætter med at køre, men skærmen "cleares" inden den vises. I praksis svarer det til at der tegnes en uendelig stor sort firkant oven på alting. Blankningen forbliver indtil der modtages en form for kode eller signal. I praksis indtil du gøre noget der skulle gøre noget på arbejderen.

Bruges typisk interaktivt til at stoppe visningen af sangtekster når sangeren synger forkert. Så blanker AVmanden, finder frem til hvor sangeren er nået til, og fortsætter

så. Når AVmanden fortsætter holder blankningen op og teksten vises som normalt.

`{revy-blank-all}`
Som `revy-blank`, men blanker alle arbejdere. Vær opmærksom på at arbejderne venter enkeltvist på at få et signal om ikke længere at være blanke.

Bruges ikke så ofte da formålet med at blanke typisk kun er at fjerne det der er forkert, sangerens tekst, og ikke alt hvad der også er på de andre arbejdere.

`{revy-unblank-all}`
Sender signal til alle arbejdere om at de ikke længere skal være blanke. Bruges typisk efter et `revy-blank-all`

`{revy-calibrate}`
Viser et kalibrerings billede. Er en simpel måde at sikre sig at alle arbejdere virker, er tændt og deres projektorer står indstillet korrekt.

5.4.3 Audio Visuelle kommandoer

`{revy-image file &optional position}`
Sender et program til den nuværende arbejder der viser et billede. Den indbyggede `image` funktion på arbejderne tegner et billede på en position en enkelt gang i et enkelt frame. For at få vist et billede i længere tid kan man bruge denne funktion der sender en stump kode der definerer en billede viser der kontinuerligt viser billedet. Den bliver ved med at tegne billedet i hvert frame indtil den afbrydes. `revy-image` sætter bare det nuværende kode til at være denne billede viser.

Se 6.1 for hvordan det præcist virker med at afvikle kode i hvert frame.

```
(revy-image "billeder/hund.png") ;; Viser et billede af en hund.
(revy-image "billeder/kat.png")  ;; Viser et billede af en kat istedet.
```

"Position" argumentet er lidt magisk og kan en hel masse

Note: TODO: Forklar noget om dette

`{revy-image-preload &rest files}`
Bruges ikke til noget.

`{revy-pdf-open file}`
Åbner en pdf fil og viser slide 0 (det første). Til sangtekster vil man typisk bruge ubertex5.2 som også sagtens kan bruges til at lave andre slideshows en bare overtekster. Til billeder og diasshow er det ofte at foretrække at bruge enkelte billeder da de nemmere kan justeres enkeltvist med "position" argumentet for hvert enkelt slide. Men ind imellem får man som AV-mand en pdf med billeder der skal vises. Til dette kan man bruge `revy-pdf-open` som definerer en pdf-viser meget lig den førnævnte billedeviser.

Vær opmærksom på at det kan tage lidt tid at åbne en stor pdf.

Note: Vær opmærksom på at revysystemet 0-indeksere slides i pdf'er. en pdf på 3 sider vil da indeholde slide 0, 1 og 2.

{revy-pdf-goto-slide slide}

Skifter hvilket slide den *nuværende* pdf-viser viser. "slide" er et 0-indekseret tal der angiver hvilket slide der skal vises.

Note: Hvad sker der hvis tallet er størrere end antallet af slides?

{revy-pdf-next}

Skifter slidet den *nuværende* pdf-viser viser til det næste

Bør generelt ikke bruges i en sketch da man ikke længere kan gå til en vilkårlig position og fortsætte. Hvis en skuespiller f.eks. hopper fra slide 5 til slide 8 i sit diasshow, kan man ikke hoppe med hvis man kun er sat op til at bruge `revy-pdf-next`. Hvis man derimod bruger `revy-pdf-goto-slide` kan man rykke direkte ned til slide 8 og vise dette.

Note: Hvad sker der hvis det nuværende slide er det sidste?

{revy-sound file &*optional* volume}

Afspiller en lyd. Lyden bliver afspillet som en sepperat komponent, så den har ikke indvirkning på anden afviklet kode/lyde. Man kan således bruge revy-sound flere gange til at afspille flere lyde på samme tid. Man kan angive en ønsket volume, et heltal mellem 0 (min) og 128 (maks). Hvis ikke lydstyrken angives spilles den på maks.

Note: Vær opmærksom på at den lydmixer der bruges lige nu er ret dårlig og forringer lydkvaliteten en hel del. Til forestillingen er dette dog ikke det store problem på grund af andet lyd/larm i salen.

Hvis man virkelig ønsker sig at en bedre lydkvalitet, f.eks. ved musikken til et dansenummer kan man istedet for `revy-sound` bruge `revy-mplayer` der udover videoer også kan afspille musik. Så mister man dog en del fleksibilitet til at stoppe/fade lyde.

{revy-stop-sounds}

Stopper alle lyde der afspilles på den nuværende arbejder. Virker lidt som `revy-abort`, men kun på lyde.

{revy-fade-sounds &*optional* duration}

Virker lidt som `revy-stop-sounds`, men istedet for at stoppe lydene brat, så fades de langsomt ud. Man kan angive en varighed i sekunder, som standard tager det 4 sekunder at fade ud.

Er et godt alternativ til `revy-stop-sounds` i mange situationer da det ofte lyder bedre.

Hvis man kalder den interaktivt spørger den efter en varighed.

{revy-mplayer file &optional x y w h}
Afspiller en video. Kører det eksterne program "mplayer" på den nuværende arbejder som afspiller videoen. Kan også afspille lydfiler.

Note: Hvordan indstiller man mplayer?

{revy-kill-mplayer}
Lukker alle instanser af "mplayer" på den nuværende arbejder. Ofte kan man bruge revy-abort istedet.

{revy-text &optional text}
Viser en tekst midt på skærmen af den nuværende arbejder.

Kan kaldes interaktivt, hvor den spørge om en tekst der så vises. Man kan bruge piletasterne til at finde tidligere tekster.

Note: Snak om Markup sproget

5.4.4 Interaktive kommandoer

{revy-create}
Opretter en ny revy. Fører dig igennem en længere "wizard" (opsætnings guide).

Note: Forklaring om hvordan denne virker

{revy-load}
Loader en allerede oprettet revy. Spørger dig om du vil indlæse en af de revyer den kender til, typisk den du arbejder på. Ellers kan du vælge "other revy..." hvorefter du kan angive den præcise placering for en revy.

{revy-mode-enter}
{revy-mode-next}
{revy-mode-point-forward}
{revy-mode-point-backward}

5.4.5 Andre

{revy-on-worker worker function &rest args}
{revy-create-workers}
{revy-send-lisp}
{revy-send-command}
{revy-shell}

{revy-shell-sync}
{revy-shell-local}
{revy-shell-local-sync}
{revy-elisp}
{revy-upload-files}
{revy-upload-files-sync}
{revy-build}
{revy-compile-tex}

6 - Worker

Note: Generelt om hvordan en worker virker

6.1 Kommandoer

{update}
Du skal forstå dette her hvis du vil lave noget som helst avanceret... Så er det ærgeligt at dokumentationen er mangelfuld, beklager...

7 - Lisp

Vectorer er ikke ens. I Emacs lisp er de konstante. I Worker Lisp er de dynamiske arrays. ak indsætte foran og bagi i O(1) tid (ammortiseret)

Lister er en slags "meta" type over consceller. En konvention! En liste er en Cons-celle hvor `car` indeholder et element i listen, og `cdr` indeholder en liste (resten). Alternativt er en liste `nil` som angiver den tomme liste eller slutningen på en liste. Der er således ikke en konkret type i sproget der hedder *list*, men man bruger det som om der var.

Quote opfører sig anderledes (Skal jeg lade være med at deep copy'e?

7.1 Hvordan programmerer man?

7.2 De forskellige lisp dele

8 LaTeX

9 - Værktøjer

9.1 Schneider

Tysk: Skrædder

Dependencies:

- Python3
- ffmpeg

Schneider er et værktøj til at skære film og billeder op til at kunne blive vist over flere projektorer.

Schneider kan på skrivende stund kun dele medier op i flere snit ved siden af hinanden og ikke over under. Det burde dog være let at rette til.

9.2 Zeitherr

Tysk: Timelord Er en bastardiseret udgave af en NTP tidsserver til at holde de forskellige maskiner synkroniseret.

9.3 Zeigen

Note: Externe:

9.4 xpdf

Overtekster køres i xpdf der åbnes manuelt med:

```
xpdf -remote ubertex -fullscreen -mattecolor black -fg black
    -bg black -papercolor black filnavn
```

Da dette er meget langt kan man istedet bruge aliaset

```
p filnavn
```

9.5 mplayer

Til at vise videoer manuelt bruges mplayer:

```
mplayer -nolirc -msglevel all=-1 -msglevel statusline=5
    -vo gl2 -autosync 30 -cache 1048576
    -cache-min 99:100 -xy 500 -geometry 49%:40% filnavn
```

eller aliaset

```
m filnavn
```

9.5.1 Positionering af mplayer

TODO: Noget om positionering af mplayer her

10 - FAQ

Ubertex tager ikke højde for pauser i comments, latex gør.

10.1 LaTeX

10.1.1 File 'overtex.sty' not found.

`overtex.sty` er en fil hvori de overtex speciffike kommandoer er defineret. `revy-manus-prepare` indsætter automatisk `\usepackage{overtex}` i overtexfilerne. Når man kalder `revy-compile-tex` specificeres automatisk hvor denne fil findes. Kaldes pdflatex manuelt ved den ikke hvor denne fil findes. Den simpleste løsning er at oversætte med et kald til `revy-compile-tex`.

Alternativt kan man enten kopiere `overtex.sty` ind i mappen hvor `.tex` filen ligger, sætte shell variablen `TEXINPUTS` til at indeholde mappen hvori `overtex.sty` ligger eller kopiere filen ind i `~/texmf/tex/latex/overtex/overtex.sty` hvor den vil være synlig for oversætteren.

11 - Problemer/løsninger

11.1 Der er lag/forsinkelser

Lag er irriterende, det opleves primært som en forsinkelse fra man har trykket på knappen til der sker noget på projektoren. Det er et problem når man skal lave overtekster.

Til DIKUs jubilæumsrevy var der ca. et sekunds forsinkelte fra jeg trykkede til at overteksterne blev vist. Jeg har ikke definitivt fundet fejlen endnu men der bliver arbejdet på det.

Her er først nogle metoder til at lokalisere hvor omtrentligt forsinkelsen opstår. Det er mere eller mindre umuligt at lave konkrete målinger så det er noget man må føle sig frem til (Super naturvidenskabelig metode!).
Følgende tager udgangspunkt i Xpdf, men gøres på samme måde med Zeigen

Prøv først at sætte systemet til at køre alt lokalt. Aka, kør både Emacs og Xpdf lokalt og der kommunikeres via ssh til `localhost`. Hvis forsinkelsen stadig er der, er fejlen enten i Emacs, i Xpdf, eller i den lokale hardware.

Mine erfaringer med Xpdf er dog at det ikke er her fejlen ligger. Prøv nu at starte Xpdf på serveren og ssh ind på denne, giv nu manuelt Xpdf ordre til at skifte slide.

Min erfaring er at det er meget tilfældigt hvad der præcist sker. Jeg oplevede at delayed forsvandt efter jeg gjorde ovenstående, også det mellem Emacs og Xpdf, men kun indtil Xpdf blev lukket.

Det virker også til at Xpdf bliver langsommer afhængigt af længden af overteksterne og ikke nødvendigvis størrelsen af pdfen.

Jeg ved ikke helt hvordan dette skal løses. Det er en af grundene til at vi vil lave vores egen fremviser (Zeigen).

12 - Eksempler

Note: Med screenshots og eksempler, både på brug af program og effekter Se det som hvad jeg ville vise til et AV-føl Videosekvens på skrift/billede

A Sådan bruges "Simple Emacs"

B En guide til Linux

cd ls cp mv

ip ssh nano

hvad er /dev/sdX og /dev/sdX1,2,3 osv? hvordan fungerer IP'er?

xmonad nævn

```
xmonad --recompile
xmonad --restart
```

C Installation af Arch Linux

Dette appendix gennemgår installationen af Arch Linux fra bunden. Arch Linux er en såkaldt "Rolling release" distribution af linux, den er ikke specielt begynder venlig, men gør hvad man beder den om, hverken mere eller mindre.

Dette afsnit er ikke et du skal bruge normalt som AV-mand, men dokumenterer processen om at sætte systemet op fra bunden af, normalt vil der forhåbentligt være nogle allerede fungerende installationer du kan låne.

Guiden er baseret på `https://wiki.archlinux.org/index.php/Installation_guide` og
`https://wiki.archlinux.org/index.php/Beginners'_guide`, og antager at du har styr på linux, se evt. Appendix B.

Note: `http://www.muktware.io/arch-linux-guide-the-always-up-to-date-arch-linux-tutorial/`
`http://www.dedoimedo.com/computers/grub-2.html`

Husk at der er "auto completion" på tab-knappen.

C.1 Live USB

Først downloades `.torrent` filen fra `https://www.archlinux.org/download/` og åbnes med dit yndlings torrent program, f.eks. `rtorrent`.

Når `.iso` filen er hentet kan denne brændes til et usbstick. `https://wiki.archlinux.org/index.php/USB_flash_installation_media`

For at lave en live USB fra et allerede eksisterende linux system sættes USB'en i, uden at mounte den (eller unmount den).

Følgende kommando sletter alt på usbsticket og laver et live USB

```
sudo dd bs=4M if=/path/to/archlinux.iso of=/dev/sdX status=progress && sync
```

Hvor `/path/to/archlinux.iso` er stien til `.iso` filen. `/dev/sd`**`X`** er stedet hvor USB'en findes, f.eks. /dev/sdb, bemærk at det er uden partitionen, så IKKE `/dev/sdb1`.

Sæt nu USBen i maskinen der skal installeres, og start op fra USB'en.

For at gendanne Live USBsticket til et "normalt" USB stick

```
dd count=1 bs=512 if/dev/zero of=/dev/sdX && sync
cfdisk /dev/sdX
```

Hvis filsystemet Ext4 ønskes:

```
mkfs.ext4 /dev/sdX1
e2label /dev/sdX1 USB_STICK
```

Eller for et klassisk Windows Fat32 system

```
mkfs.vfat -F32 /dev/sdX1
dosfslabel /dev/sdX1 USB_STICK
```

For at lave et Fat32 system kræver det at `dosfstools` er installeret. cfdisk bruges til at lave en partition på USB'en, som findes bå `/dev/sdX1`, `USB_STICK` er navnet der ønskes på USB'sticket.

C.2 Installation af styresystem

Hvis man ønsker et andet tastatur layout det gøres med f.eks. `loadkeys colemak` hvis man ønsker at slå bip-lyden fra kan man gøre det med `rmmod pcspkr`

```
timedatectl status
```

Vær sikker på at tiden er indstillet korrekt, ovenstående skal sige at **Universal time** er korrekt, din lokale tidszone kan indstilles senere. Hvis tiden ikke passer ændres den i BIOS'en inden boot.

Note: Brug ntp? `timedatectl set-ntp true`

C.2.1 Opret forbindelse til internettet

Hvis det er et helt almindeligt trådet netværk virker det muligvis uden problemmer Hvis maskinen er på et seperat netværk uden internet, men med en gateway som brugt til den normalle revyopsætning gøres følgende:

```
ip link show
ip link set enpXsX up
ip addr add 192.168.0.XXX/24 dev enpXsX
ip route add default via 192.168.0.YYY
echo nameserver 8.8.8.8 > /etc/resolv.conf
```

`XXX` er ip'en til maskinen, `YYY` er ip'en til gateway'en

Note: Henvis til gateway

Hvis det er et almindeligt trådløst netværk benyttes `wifi-menu` eller `wpa_supplicant` for mere avancerede opsætninger

Test om der er internet:

```
ping google.com
```

tryk Ctrl-c for at stoppe.

C.2.2 Opret partitioner

Hele situationen omkring UEFI vs. BIOS, GPT vs. MBR, kombineret med bootloaderen (grub) og hardwareunderstøttelse er noget gøjl. Det er et virvar af forskellige systemer der ikke altid har lyst til at samarbejde, ikke fortæller hvorfor det ikke virker, og mangler dokumentation om hvordan man får det til at virke. I en periode, da det hele var nyt, og denne guide blev skrevet i første omgang, var dokumentationen nærmest ikke eksisterende og meget af det her er opdaget ved manuel afprøvning.

De næste par afsnit handler om forskellige måder jeg har prøvet at sætte systemer op på. Oprindeligt brugte jeg GPT-BIOS, da det var hvad jeg endelig fik til at virke, men prøv dig frem.

C.2.3 Opret partitioner (MBR-BIOS)

```
fdisk /dev/sda
```

med `parted -l /dev/sda` kan man se om det er GPT eller ej (også kaldet msdos) tryk `o` for at skifte type til MBR. Lav en swap partition og root + evt. home partition sæt resten op som et GPT-BIOS system

C.2.4 Opret partitioner (GPT-BIOS)

Følgende beskriver hvordan jeg plejer/plejede at oprette partitioner, se det alternative afsnit C.2.5 hvis et UEFI system ønskes. Jeg tivler på at dette er den korrekte måde at gøre det på, men det virker på ældre maskiner.

`lsblk` viser alle partitioner på alle diske, alternativt brug `fdisk -l`

Note: Følgende sletter alt indhold på harddisken og alle eksisterende partitioner

Man kan slette alt indholdet på en disk med

```
sgdisk --zap-all /dev/sdX
```

```
cgdisk /dev/sdX
```

Formententligt `sda`

Tryk enter. (til alt "brok")

`Delete` alle partitoner.

Lav plads til GPT partition *Dette giver plads til en partions tabel til grub*

1. `New`
2. Tryk enter, first sector skal bare være default.

3. `1M` enter - *tidligere 1007KiB*
4. `ef02` enter
5. Tryk enter, der behøver ikke at være et navn

Lav swap *Dette er næppe nødvendigt, men jeg laver den af gammel vane. Alternativt kunne man lave en swap fil, men en partion er simplest*

1. Tryk ned. (til det store område med free space.)
2. `New`
3. Tryk enter, default er fint, formententligt 2048.
4. `3G` mindre swap kan også vælges (eller udelades).
5. `8200` swap Hex koden.
6. tryk enter.

Lav en partition Hvis flere seperate partitioner ønskes, f.eks. opdelt `/` og `/home` oprettes de her.

1. Tryk ned.
2. `New`
3. enter
4. enter
5. enter
6. enter

Vælg `Write`, skriv `yes`, og afslut.

Nu kan der stå at den gamle partitionstabel stadig er i brug. I så fald genstart og udfør alle trinene inden `cgdisk` igen.

Når du er klar skal vi så lave nogle filsystemer. For at få overblik over partitionerne:

```
lsblk
```

```
mkfs.ext4 /dev/sda3
mkswap /dev/sda2
swapon /dev/sda2
```

(Hvis flere partioner blev oprettet til home og root, så gentag øverste linje for disse partioner)

Ignorer `sda1` indtil videre.

```
mount /dev/sda3 /mnt
```

Hvis du skulle have lavet en seperat partition til home så lav en mappe `mkdir /mnt/home` og mount home partitionen der `mount /dev/sdaX /mnt/home`.

C.2.5 (GPT-UEFI)

Dette er en lidt simplere gennemgang af at sætte et GPT-UEFI system op. Dette er den nye måde at gøre tingene på som understøttes af moderne hardware.

`lsblk` viser alle partitioner på alle diske, alternativt brug `fdisk -l`

Note: Følgende sletter alt indhold på harddisken og alle eksisterende partitioner

Man kan slette alt indholdet på en disk med

```
sgdisk --zap-all /dev/sdX
```

```
cgdisk /dev/sdX
```

Formententligt `sda`

Tryk enter. (til alt "brok")

`Delete` alle partitoner.

Lav plads til GPT partition *Dette giver plads til en EFI partitionstabel*

1. `New`
2. Tryk enter, first sector skal bare være default.
3. `512M` enter
4. `ef00` enter
5. Tryk enter, der behøver ikke at være et navn

Lav swap *Dette er næppe nødvendigt, men jeg laver den af gammel vane. Alternativt kunne man lave en swap fil, men en partion er simplest*

1. Tryk ned. (til det store område med free space.)
2. `New`
3. Tryk enter, default er fint, formententligt 2048.
4. `3G` mindre swap kan også vælges (eller udelades).
5. `8200` swap Hex koden.
6. tryk enter.

Lav en partition Hvis flere seperate partitioner ønskes, f.eks. opdelt `/` og `/home` oprettes de her.

1. Tryk ned.

2. `New`
3. enter
4. enter
5. enter
6. enter

Vælg `Write`, skriv `yes`, og afslut.

Nu kan der stå at den gamle partitionstabel stadig er i brug. I så fald genstart og udfør alle trinene inden `cgdisk` igen.

Når du er klar skal vi så lave nogle filsystemer. For at få overblik over partitionerne:

```
lsblk
```

```
mkfs.fat -F32 /dev/sda1
mkfs.ext4 /dev/sda3
mkswap /dev/sda2
swapon /dev/sda2
```

(Hvis flere partioner blev oprettet til home og root, så gentag øverste linje for disse partioner)

```
mount /dev/sda3 /mnt
mkdir /mnt/boot
mount /dev/sda1 /mnt/boot
```

Hvis du skulle have lavet en seperat partition til home så lav en mappe `mkdir /mnt/home` og mount home partitionen der `mount /dev/sdaX /mnt/home`.

Du kan bruge `lsblk` til at få overblik over partitionerne, og hvor de er mounted.

C.2.6 Installation

Sørg for at være på nettet da vi nu skal hente pakker ned til styresystemet.

```
pacstrap -i /mnt base base-devel
```

Tryk enter til spørgsmål.

```
genfstab -U -p /mnt >> /mnt/etc/fstab
arch-chroot /mnt /bin/bash
nano /etc/locale.gen
```

Fjern kommenteringen til linjerne *#da_DK.UTF-8 UTF-8* og*#en_US.UTF-8 UTF-8*

Note: Henvis til brug af nano

```
locale-gen
echo LANG=en_US.UTF-8 > /etc/locale.conf
export LANG=en_US.UTF-8
ln -s /usr/share/zoneinfo/Europe/Copenhagen /etc/localtime
hwclock --systohc --utc
echo XXX > /etc/hostname
```

hvor XXX er navnet til maskinen

Note: Hvis du hellere vil arbejde med at netværksinterfacene hedder `eth0` osv. istedet for enpXsX: `touch /etc/udev/rules.d/80-net-setup-link.rules`

Note: ?

kopieret fra noter: Brug netctl til automatisk at gå på netværk (Ret i hovrdan man bruger deet i resten af dokumentationen)

```
curl "https://raw/githubusercontent.com/Pilen/ubertex/master/linux/revynet" >
 /etc/netctl/revynet"
nano /etc/netctl/revynet
```

Og så skal det gerne matche følgende:

```
Description='Default static network configuration for the AV-setup'
Interface=eth0
Connection=ethernet
IP=static
Address=('192.168.0.XXX/24')
Gateway='192.168.0.YYY'
DNS='8.8.8.8'
SkipNoCarrier=yes
```

Hvor XXX erstates med den lokale IP, og YYY med gateway'ens.

```
netctl enable revynet
```

Hvis man ændrer i configurationen, f.eks. IP'en:

```
netctl reanable revynet
netctl restart revynet
```

Hvis maskinen skal kunne gå på trådløst netværk efterfølgende;

```
pacman -S iw wpa_supplicant dialog
```

```
mkinitcpio -p linux
```

```
passwd
```

Skriv løsn til root (f.eks. `hamsterroot`).

Note: Du kan med fordel oprette din bruger her

Note: Du kan med fordel sætte sudo op her

Note: Du kan med fordel installere sshd her

For et GPT-BIOS system:

```
pacman -S grub
grub-install --target=i386-pc --recheck --debug /dev/sda
grub-mkconfig -o /boot/grub/grub.cfg
```

For et GPT-UEFI system:

```
mkdir /boot/efi
pacman -S grub efibootmgr
grub-install --target=x86_64-efi --efi-directory=esp --bootloader-id=grub
grub-mkconfig -o /boot/grub/grub.cfg
```

Note: Den vil klage og sige: `efibootmgr: EFI variables are not supported on this system`, ignorer dette?

Vi er nu klar til at genstarte op i det nyinstallerede system:

```
exit
umount -R /mnt
shutdown -h now
```

Note: ?

Note: Hvis grub-mkconfig fejler:

```
nano /etc/default/grub
...
...
#fix broken grub.cfg gen
GRUB_DISABLE_SUBMENU = y
```

Hiv usbstikket ud.

Tænd datamaten igen. Hvis du som mig til jubilæumsrevyen installerede systemet på en harddisk i en anden datamat en dens egen, kan det være at ramdisken fejler, i grub vælges da fallback løsningen og du kalder `mkinitcpio -p linux` for at skabe et nyt korrekt image.

C.3 Opsætning

C.3.1 Bruger

Note: Hvis en bruger og sudo er sat op kan man logge ind som dem, ellers så log ind som root

Log ind som root

Note: Kom på nettet igen

```
ip link show
```

For at se netværks interfacet

```
useradd -m -s /bin/bash revy
```

```
passwd revy
```

Giv revy et løsn.

Sudo er allerede installeret via base-devel, du kan læse mere om sudo i Appendix B.

```
visudo
```

Tryk pil ned til du finder linjen

```
root ALL=(ALL) ALL
```

placer cursoren under denne og tryk `i` tast `revy ALL=(ALL) ALL` tryk enter, tryk escape tryk :wq enter.

revy kan nu sudo'e.

```
pacman -S openssh
```

Note: Det er nu default at `PermitRootLogin` er sat til `no`, så det er ikke længere nødvendigt at ændre det i `/etc/ssh/sshd_config`

```
systemctl enable sshd.service
systemctl start sshd
```

Nu kan vi ssh'e ind fra vores lokale maskine `ssh revy@192.168.0.XXX`. *Læs evt. afsnittet om ssh-nøgler.*

Nu kan vi vælge at køre kommandoerne via en ssh forbindelse. Du kan også genstarte og logge ind som almindelig bruger istedet, flere af kommandoerne kræver da sudo.

C.3.2 Grafisk system

Nu skal vi installere et grafisk interface

```
pacman -S xorg-server xorg-server-utils xorg-xinit
```

Den spørger nu hvilken udbyder af `libgl` der ønskes, default (1 mesa-libgl) er formententligt et fint valg (medmindre andet ønskes). (tryk 1 efterfulgt af enter)

Den spørger så om hilken udbyder af `xf86-input-driver` der ønskes, 1: `evdev` eller 2: `libinput`. Libinput er et nyere wayland projekt der også virker til xorg. Begge er tilsyneladende fine, jeg valgte evdev da det er det "klassiske" valg.

Der skal formententligt installeres nogle grafik drivere, læs mere her `https://wiki.archlinux.org/index.php/Xorg#Driver_installation` `fbdev` og `vesa` er nogle udemærkede open source fallback drivere hvis ikke der findes et grafikkort (bemærk at det fører til software rendering). Hvis det er et intel kort, er intel driveren et godt bud, for nvidia er nouveau driverne et godt open source bud

Note: mesa og mesa-libgl er installeret af xorg-server

```
pacman -S xf86-video-fbdev xf86-video-vesa xf86-video-intel
```

Derudover er det en god idé med Hardware video acceleration

```
pacman -S libav-intel-driver libav-mesa-driver
```

Som windowmanager bruger vi xmonad, en simpel "tiling" windowmanager med minimale vinduesdekorationer, kan styres med tastaturet og fordi jeg kender den. Se Appendix B for mere om brugen.

```
pacman -S xmonad xmonad-cotrib
```

Vi skal oprette en `.xinitrc`, den kan hentes sådan

```
curl "https://raw/githubusercontent.com/Pilen/ubertex/master/linux/.xinitrc" > .xinitrc
chown revy:revy .xinitrc
nano .xinitrc
```

Indholdet skal da være:

```
xdotool mousemove 10000 10000                          # Flyt cursor til meget langt væk
xsetroot -cursor .emptycursor.xbm .emptycursor.xbm     # slå cursor fra
xset -b                                                # fjern bell/bip lyd
xset -dpms                                             # Slå energi besparelse for skærmen
xset s off                                             # Slå pauseskærm fra
exec xmonad                                            # Start xmonad/windowmanager
```

Den usynlige cursor hentes med:

```
curl "https://raw/githubusercontent.com/Pilen/ubertex/master/linux/.emptycursor.xbm" > .em
```

```
mkdir .xmonad
curl "https://raw/githubusercontent.com/Pilen/ubertex/master/linux/xmonad.hs" > .xmonad/xm
chown -R revy:revy .xmonad
```

```
import XMonad
import XMonad.Layout.NoBorders

main = xmonad $ defaultConfig {
  terminal = "urxvt",
  borderWidth = 1,
  normalBorderColor  = "#000000",
  focusedBorderColor = "#000000",
  modMask = mod4Mask, -- Set win as mod key
  layoutHook = smartBorders $ layoutHook defaultConfig
  }
```

Note: Skal borderWidth være 0?

```
xmonad --recompile
```

Åben `.bashrc` og tilføj i bunden:

```
if [[ -z $DISPLAY && $(tty) = /dev/tty1 ]]; then
    exec startx
fi
```

Du kan evt. også tilføje følgende alias over linjerne:

```
alias d='export DISPLAY=:0'
```

For at systemet automatisk kan logge ind ved opstart kan man gøre følgende

```
curl "https://raw/githubusercontent.com/Pilen/ubertex/master/linux/autologin.conf" > /etc/s
nano /etc/systemd/system/getty@tty1.service.d/autologin.conf
```

indhold af `autologin.conf`:

```
[Service]
ExecStart=
ExecStart=-/usr/bin/agetty --autologin revy --noclear %I 38400 linux
```

Note: I noterne står det som `curl "https://raw/githubusercontent.com/Pilen/ubertex/master/linux` Outputet fra curl kan skjule forespørgslen efter løsen, så hvis den hænger er det nok derfor

C.3.3 Pakker

```
pacman -S alsa-utils rxvt-unicode dmenu xdotool feh mplayer ttf-dejavu screen htop rsync mu
```

D Worker-protokol

Note: Her burde følge en introduction af protokollen mellem master og worker.