Schedule

10.00  Welcome and dwm, arg
10.45  st, 20h
11.30  musl, nsz
12.15  sta.li, arg
14.00  runit and ignite, chneukirchen
14:45  costs of abstraction, nsz
15:30  suckless web ideas, 20h
16:30  next generation and suckless ev, arg
Welcome

- First slcon ever
- Thanks to chneukirchen and all contributors

Some facts

- Founded in 2006, previously wmii.de
- Initially dwm, dmenu, now various projects
- Philosophy is spreading
- IRC channel lurker base grew from 10 to 140
- Approx. 60k downloads of dwm-6.0
- Userbase probably around 10k
- It was time to have the first slcon
Future of dwm

- Almost finished project
- However:
  - X changes
  - Users come up with new ideas
  - New approaches appear (Wayland/Mir)
  - Architecture and design is always questioned/revised (suckless principle)
- Recently: has the dwm development stopped?
  -> No, only suspended for personal reasons
dwm sloc evolution during time
So what are the next steps?
Next steps in dwm development

- libsl separation and re-use
  - Drawing code API-abstraction
  - Utility code abstraction
- Drawing code API allows different implementations
  - X11/Xft
  - X11/cairo
  - X11/whatever
  - Wayland? (probably not)
- Reuser of libsl in other projects like dmenu and hopefully st (needs to be discussed)
Plan

- 6.1 around Oct 2013
  - All next steps included
- 6.2 in 2014 (probably the final release)
  - remember, dwm is finished
- Thanks
Future of sta.li

- Current state: still vaporware
- Contradicting approaches and plans
  - starch
  - sabotage
  - uclibc
  - musl
  - android
  - installer vs out of the box/netbootable
  - different filesystem layout wishes
  - different user/group handling wishes
  - single user vs multi user
  - nldev/mdev
  - etc
sta.li fs

/bin - all executables
/boot - all boot files
/dev - devices
/etc - system configuration
/svc - services
/home - user directories
/root - the root home
/var - spool, run, log, cache
/share - man pages, locales, dependencies
/devel - development environment
/devel/include
/devel/lib
/devel/src
My take on it

- To me the sta.li/ page still explains what sta.li should be
- I'm investigating android as base after looking into the options, but with musl instead of bionic
- Reasons
  - Google almost killed the traditional /etc and user/group handling
  - It is not designed for servers (no real sta.li goal from my perspective)
  - The core of the android build system is actually sane (most of it is actually insane, though)
Plan

● Tonight: git.sta.li launch
● android based prototype until Aug 2013
● Problem: lack of time
  ○ See my later talk in the afternoon how to tackle this
● We need to collaborate, it is impossible to achieve with 1 man power
  -> Requires agreeing what to do and how

● Thanks
Next generation

● How to spread the suckless idea further?

● What about the new contributors?
  ○ So far there are several kinds of contributors
    ■ long time contributors
    ■ short time contributors
    ■ sporadic contributors

● How to further develop the suckless philosophy?

● How to change the decision making process?
Idea: suckless e.V.

- Formal legal entity
- Possibility to hire students working on projects
  - GSOC rejected us 3 times, now we do it ourselves
- Easier to organize things and to find sponsors
- Easier to build up the suckless certification system ;)
- You tell me what else would be possible...
Closing summary