

# GNU Screen

Abram Hindle

## 1. INTRODUCTION

In this paper we plan to discuss the open source community that surrounds GNU Screen [20]. This is both an overview of a community and its history as well as an experience report which documents an attempt to create and submit a patch to the GNU Screen project.

This study took place from January 2005 to April 2005. As of this study GNU Screen was an 18 year old project with the latest release of 4.0.2 and a size of near raw 44KLOC.

## 2. GNU Screen

GNU Screen is a very mature free open source project. GNU Screen was created in 1987, it was meant to provide a “window manager” to those users who didn’t even have windows.

GNU Screen is a virtual terminal emulator. Specifically Screen emulates the VT100 terminal and allows a user to open multiple interactive sessions and switch between these sessions. It allows users to open multiple terminal sessions within one terminal, much like a tabbed terminal emulator. GNU Screen also users to disconnect and reconnect to their terminal sessions. This allows both interactive and noninteractive programs to run safely in the background and to allow the user to run large jobs to come back later to monitor them. Thus screen allows users to have multiple windows, to switch between windows, create new windows and to attach, detach and reattach to and from screen sessions consisting of multiple windows. This allows for effective multitasking of interactive and non-interactive console based applications. This is especially useful if you are connecting to a console via an unstable link [13].

Screen runs on many UNIX based platforms from OSF/1 to Linux to BSD to Solaris. Screen is implemented in C and can be compiled by many compilers on various platforms.

### 2.1 Motivation

Oliver Laumann, the creator screen said during an email interview [13]:

There were basically two things that I wanted screen to accomplish.

First, when screen was invented, we were still working on 24x80 ASCII terminals; workstations with bit-mapped graphics screens and window managers were virtually non-existent at that time. Screen was written to let the user switch between several different tasks on the ASCII terminal. For example, screen should enable the user to run a compiler in one virtual console and then create another virtual console to write an e-mail without having to interrupt the compiler.

Second, we often started a UNIX command that would run several hours (e.g. a compiler) and then went home while the command was still running. Screen should enable users to connect to their office computer via telephone modem from home and then re-attach to the command they started when they were still in the office. This is what “screen -t” does.

Thus the motivations of screen are to provide a virtual console where you can view the output of and interact with many programs such as an Email program, an Internet Relay Chat client and a compilation job.

I personally use screen to monitor multiple long jobs and to persistently run an email client, PINE, and a IRC client such that I can disconnect freely or use poor network connections to temporarily monitor programs.

It seems that based on the initial release of screen that multitasking was more important than the reattaching features. It wasn’t until screen 2.0 in late 1987 that screen allowed users to detach and reattach to a disconnected screen session. Of course screen 1.0 and 2.0 started during the same year.

### 2.2 License

In 1987 Screen started off with an academic license similar to the MIT license. This license came from screen 1.1i:

screen-1.0	1987	
screen-1.0g-2	6-Sep-1987	
screen-2.1b	26-Jul-1991	
screen-3.0.1	08-07-1991	(CVS)
screen-3.1.0	09-09-1991	(CVS)
screen-3.2.0	31-01-1992	(CVS)
screen-3.3.0	19-03-1993	(CVS)
screen-3.4.0	16-06-1993	(CVS)
screen-3.4.0	16-06-1993	(CVS)
screen-3.5.0	22-07-1993	(CVS)
screen-3.6.0	20-12-1994	(CVS)
screen-3.6.0	16-Jan-1995	
screen-3.7.1	01-Oct-1996	368k
screen-3.7.2	18-Dec-1996	372k
screen-3.7.4	17-Jul-1997	379k
screen-3.7.6	16-Dec-1998	382k
screen-3.9.4	16-Aug-1999	488k
screen-3.9.8	01-Sep-2000	493k
screen-3.9.9	16-Aug-2001	513k
screen-3.9.10	04-Sep-2001	509k
screen-3.9.11	14-Feb-2002	705k
screen-3.9.15	13-Mar-2003	810k
screen-4.0.2	27-Jan-2004	821k

**Table 1: Screen Version Releases**

University of Berlin. Not derived from licensed software.

Permission is granted to freely use, copy, modify, and redistribute this software, provided that no attempt is made to gain profit from it, the author is not construed to be liable for any results of using the software, alterations are clearly marked as such, and this notice is not modified.

Around 1991 Juergen and Michael Schroeder were given official control of screen by Laumann. Laumann says he gave control away because he did not have time to work on screen anymore [13]. It was apparent that Laumann did not want to add certain features to screen that the community seemed to want [12]. Juergen and Michael Schroeder worked on a parallel fork of screen, iScreen [18], which had many more features than Laumann's screen. This fork was based upon work done by Wayne Davison who created a custom screen 2.3PR7 with more features [21]. As part of the transfer Screen would be licensed under the GPL license 1. Although as of July 26, 1991, Screen version 2.1b [7] was still released by Laumann on comp.unix.sources under the MIT-Style license. This is strange as Laumann says that "screen 2.1a is distributed under the terms of the GNU copyleft" [11]. iScreen version 3.1.1 [19] was the first version by Juergen Weigert to be released under the GPL (officially). Later versions of screen were licensed under the GPL Version 2. A history of screen releases is documented in table 1.

Versions before Screen 3.0.0 were maintained and numbered mostly by Oliver Laumann. used a simple and arbitrary revision numbering scheme. The first number was the major release (1., 2.) the second number was a minor release (1.1). Minor releases added new features and fix new bugs. Versioning is further muddled by addition of letters to end

of version, which seem to indicate pre-release status or minor bug fixes (it is not clear). Forks created by other users such as Wayne Davison and his 2.3PR7 release muddled the versioning of Screen 2.0

GNU Screen versions 3.0.0 to 4.0.2 use a Major Release, Minor Release and bug fix release version notation. Note there are two different versioning systems, 1 used by CVS internally by Juergen and Michael and one used externally for external releases. Both internal and external version numbers have corresponding Major and Minor release numbers. They differ in the bugfix number. For instance there was a 3.9.98 in CVS as indicated by patchlevel.h but the last public bugfix release was 3.9.15, The last number simply counts the number of bug fix releases publicly added. Where as in the patchlevel.h the developers indicate each time a patch comes in or they make a reasonable change the bug fix release number increases.

## 2.3 Community

Laumann reports that early in the life of screen there was much community involvement in screen [13]. He received many questions and a quite a few patches. USENET posts can be found from other users who were going to submit a patch or submitted a patch to Screen [10]. Many patches were added to support new architectures.

Originally screen was distributed via the newsgroup comp.sources.unix while being mirrored on a ftp site belonging to the AI Lab at MIT (ftp://pre.ai.mit.edu/pub/gnu/screen). Discussion regarding screen occurred on comp.sources.d, comp.sources.bugs, gnu.utils.bug and various UNIX vendor newsgroups before the mailing lists were setup in 2000.

In April of 2003, screen was registered on the FSF's GNU Maintenance and tracking site Savannah [4].

A mailing list community was at Yahoo Groups on the gnu-screen mailing list (referred to as comp.unix.screen) from May 2000 to May 2004 [6]. The user community then moved to the gnu.screen.user mailing list hosted on GNU Savannah [4].

The screen community seems reasonably active for a large stable project. The user community seems to consist of system administrators and UNIX users. Posters seem quite bursty, they appear for a short period of time and then stop posting. Posters probably join the list because they have one problem, once that problem is solved they no longer need the mailing list. Figure 1 plots the posts per unique poster. Most of the messages seem to be configuration, usage or bug related. The two mailing lists had in total 485 unique posters (22 posters were found in both mailing lists) from July 23, 2002 to March 28, 2005 (there are some duplicate posters as well). On Savannah from April 5th 2003 to March 28th 2005 there were 134 unique posters on gnu.screen.user who made 527 messages. There were 373 unique posters on comp.unix.screen from July 23, 2002 to December 1, 2004 who made 2108 non spam messages out of 2118 messages.

In patchlevel.h there were 20 different patches or bugs at-

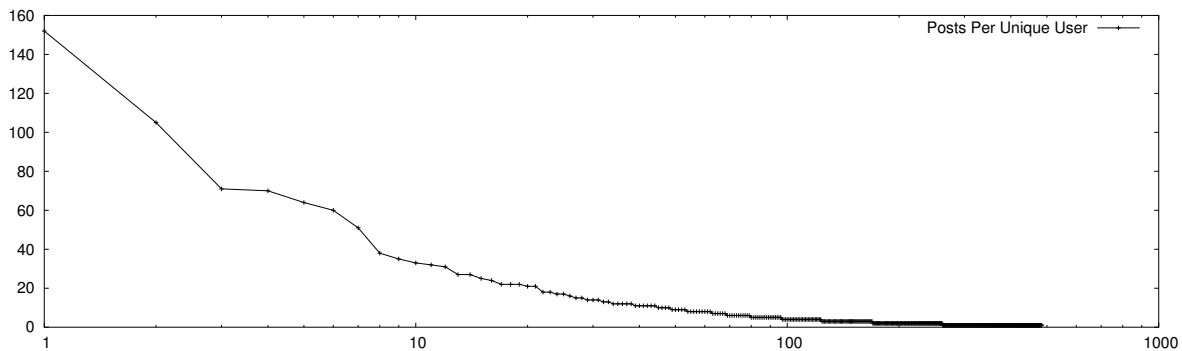


Figure 1: Number of posts per unique user

tributed to other users other than Weigert and Schroeder. In the Changelog there were 4 changes attributed to other users but the Changelog was not used past Screen 3.5. NEWS files were used for each major screen release after that. The manual [21] lists 26 contributors.

Thus screen has a core group of 2 long term maintainers and 2 other project Managers. Surrounding the group is more than 20 developers who have submitted 1 or more patches to screen. Table 2 describes the roles of various maintainers and authors of the screen project.

Community contributions come in the way of patches, documentation and build system changes. Contributions are submitted directly to the screen maintainers at `screen@uni-erlangen.de`. Contributions can also be submitted via GNU Savannah bug tracker (although one has to prod the developers to notice) or via the mailing list. Pressure is often applied to the developers through the mailing list [2].

Contributions seem to be strictly controlled by the reigning maintainers. Oliver Laumann refused patches from other developers because he said he was working on a patch which did the same thing [8, 9]. Laumann suggests that he accepted many patches via email. The current contributions are added via CVS to the screen project by Jeurgen Weigert and Michael Schroeder. They choose what belongs in screen and what doesn't. Most patches seems to be eventually accepted.

## 2.4 Development

Currently the Screen users community uses GNU Savannah [4] as its collaboration tool.

Originally screen sources were distributed via USENET using the `comp.unix.sources` newsgroup and via ftp at `ftp://pre-ai.mit.edu/pub/gnu/screen`. Now screen is primarily distributed via ftp on the GNU Savannah site [4]. Screen is also packaged and distributed with many Linux distributions such as Debian, Ubuntu and Fedora Core.

The two main maintainers of Screen are Jeurgen Weigert and Michael Schroeder. Currently source trees are publicly available in FTP in tarballs (.tar.gz). The maintainers have a private CVS tree for the screen source (Started in 1993 with Screen 3.04). CVS is currently not publicly used but

the maintainers plan to use some form of Version Control in the future [16].

### 2.4.1 Tools and Libraries

The software configuration system by GNU Screen is GNU autoconf. Autoconf allows screen to generate makefiles based on the configuration information extracted from the environment and the user. GCC (GNU Compiler Collection) is the default compiler of Screen, of course Screen provides support for other environments from OSF/1 to HP-UX thus it supports a multitude of Unixes. GNU Make is commonly used to run the Makefiles created by autoconf to compile, assemble, link and install screen.

External libraries that screen uses are PAM (Pluggable Authentication Modules). PAM allows system administrators to configure authentication controls on their systems. NCurses is a terminal library that enables a program to extract environment information about the terminal and display such things as highlights colors, paddings and process Unicode.

Distribution tools used by the screen project include Savannah (the GNU Savannah System), FTP, NTP (USENET) and mailing lists.

## 2.5 Project Measurements

As of Screen 4.0.2 screen contains 32 .c files and 18 .h files. There are 4 main sub directories which do not contain code that is compiled into Screen executable:

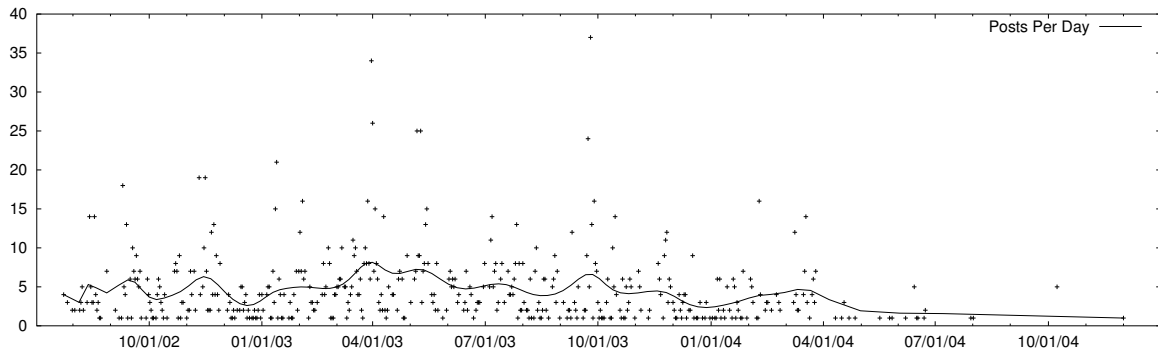
- ./ - the main directory which contains the source code.
- doc - documentation directory (contains man and info pages).
- etc - braille and screen resource data.
- terminfo - terminal information data.
- utf8encodings - lists of utf8 encoding data.

Screen documentation consists of a README, a Changelog (currently not used), NEWS (per version Changelog), info and man pages.

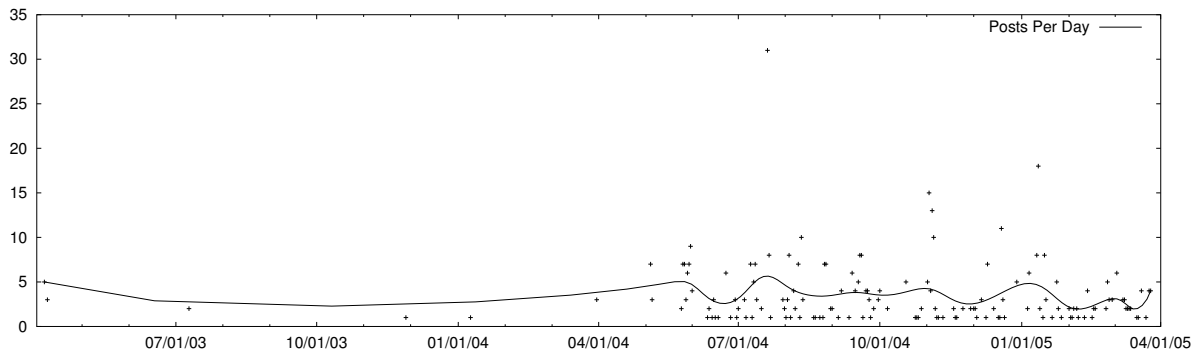
Out of 44 RAW KLOCS 4923 LOCS are dedicated to preprocessor directives (specifically lines that start with #). These

Name	Username	Job
Juergen Weigert	jnweiger	project admin
Michael Schroeder	mschroeder	project admin
Quasar Jarosz	MasterRa	support, bug, patch, news tracker
Tom von Schwerdtner	tvon	project admin
Screen Dummy User	screen	(Savannah)
Wayne Davison		Created a custom screen 2.3PR7
Oliver Laumann		Original Author (no longer involved)

**Table 2: Screen Maintainers on Savannah**



**Figure 2: Posts per day on comp.unix.screen**



**Figure 3: Posts per day on gnu.screen**

lines include `#ifdef`, `#include`, `#endif`, `#define`, `#ifndef`, `#else`, `#if`, `#define`, , etc. There were 863 `#define` lines , 197 `#include` lines , 884 `#ifdef` lines 104 `#ifndef` lines , and 363 `#if` lines. This suggests that the preprocessor is heavily used to determine optional code. Optional code implies support for multiple architectures, this is quite evident as 44% of all preprocessors lines are dedicated to defines and conditionals. Even more startling is that 10% of all KLOCs are dedicated to the preprocessor.

### 3. CONTRIBUTION

I use screen everyday. I use it to switch between my email, my IRC client and various shells. I use it to run long jobs (like music classification) so that I can come back later and see the current status. screen is useful to execute large jobs that require some interactive use or at least require almost full control over the console (you cannot run them in the background).

One of my problems with screen lately was that I would SSH into screen and ask it to reconnect to my email session. Sometimes it would just hang. If I left it, it would hang for hours. This would mean I'd just kill off my IRC client and mail client before I could start them again, possibly meaning I could lose what I was working on at the time.

Effectively my contribution was going to consist of me scratching my own itch [14] .

#### 3.1 Contribution Submission

My contribution to the screen project was to test, find and fix the bug which caused Screen to hang when we wanted to reattach.

I found a similar bug report related to the bug I was looking for: "bug #11610 overview: XON (Ctrl-S) halts screen 4.0.2" [5]. My problem was that screen was freezing when SSH disconnects or even sometimes while I was using screen. I thought that this bug could be related.

What the bug was referring to was when scroll lock or CTRL-S was hit the user couldn't unlock the terminal by pressing CTRL-Q or scroll lock again. Scroll lock was used on older style terminals to pause the screen when lots of text was scrolling by.

The author of the bug report claimed that he could not reconnect to a scroll locked screen session. The session must've been waiting for a scroll lock release (CTRL-Q) command. When the author attempted to reconnect to the screen session he couldn't ("screen -D -A"). There was some discussion on the Debian list [3] that way to fix this bug was simply to kill off the client screen processes and even the ssh sessions and then reconnect.

My problem was specifically similar because I could not reconnect to the screen session as well. I then compiled screen with debugging options enabled with the compiler. Then I tried to replicate the situation. The best way to replicate the situation seemed to be starting a screen session and outputting a lot of text to the screen, then to close the terminal window. I would then SSH into the machine and attempt to reconnect to my lost screen session. It would hang. But now

that I had debugging symbols embedded in the executable I could connect GDB to screen via the process identifier of the screen daemon. Once I was in GDB (GNU Debugger) I could simply tell the process to continue and then my client session would be connected. It seemed like I had to "touch" the daemon to remind it to read data from the client .

So by using the debugger on the clients and the daemon I determined it was the daemon blocking the clients. After further debugging I found 2 locations where it would hang. One spot was in the TTY flush routines which would write bytes out to the client and the other spot was in the setTTY routine which called NCurses. It is not up to NCurses to do any sort of timeouts and there was only one other timeout found in the screen code. The problem was that these two areas were either reading or writing to a file handle which was blocked and probably indefinitely blocked (SSH didn't seem to detect disconnections very well).

My solution was to use an alarm signal and `sigsetjmp` to give a 20 second timeout to sessions during `Flush` and `setTTY`. I then tested the solution for a bit and it did work. If there was such a problem I would attempt to reconnect, wait 20 seconds and then I'd be reconnected. The scroll lock problem was solved as well, if you disconnect and reconnect, the session would be restored.

#### 3.2 Submission to Screen

After cleaning up and testing my patch I used diff to produce a patch. I created a comment for bug #11610 on the Screen Savannah site. Attached to my comment was my patch and a some lines captured from GDB while debugging. This was on January 27th, 2005.

It wasn't until February 25th when I messaged the list (`gnu-screen.user`) asking questions about patches till the maintainers (specifically Michael Schroeder [16]) noticed my patch.

On February 11th someone on Savannah confirmed the bug as well. On February 26th someone applied my patch and suggested that a new version of Screen should be released since he had experienced this bug before and it cost him "an immense amount of productivity" [5] . On the 27th someone cross-posted to Savannah and the Debian bug tracker regarding 2 patches existing for the one. The Debian [3] bug fix was to change some NCurses flow control settings.

On March 25th 2005 Eric Smith wrote to the mailing talking about the reattach problem (reconnecting to a session) [17]. I had some correspondence with some users who report that my patch worked for them. I asked them to combine it with the Debian patch.

It should be noted that a patch to screen was submitted, "bug #11021 overview: `:exec !.` and `Report()`", by Stephane Chazelas [1, 2] on November 16th, 2004 was not commented by the maintainers even after Chazelas posted to the list in on January 26th , 2005. I mention this as my patch has yet to be officially rejected or accepted and it not as old as Chazelas's patch.

### 4. FUTURE OF SCREEN

Future directions for screen include more mouse support and some discussion about integration with tabbed graphical GUIs like tabbed terminals such as Gnome-Terminal [15].

I would recommend that the screen project move to a publicly accessible CVS repository and come up with a publicly known and coherent way that contributions are dealt with. Make the structure of how things are added clear and where to send patches. Michael Schroeder suggested that screen will have a publicly accessible CVS soon [16].

Since 10% of the project consists of preprocessor directives I'd recommend that the project should undergo an audit of this code to see how much is actually necessary to support current platforms. This is a hard problem due to the fact that the maintainers probably do not use all the architectures that are supported. This code might provide problems in the long run.

## 5. CONCLUSION

GNU Screen was started in 1987 by Oliver Laumann is still used and maintained to this day by Jeurgen Weigert and Michael Schroeder at the University of Erlangen.

The GNU Screen project is a very stable and mature project that is used by many people. Particularly those who use UNIX remotely or before X11 and window managers were really available.

## 6. REFERENCES

- [1] S. Chazelas. bug #11021 overview: :exec !. and report(). [http://savannah.gnu.org/bugs/?func=detailitem&item\\_id=11021](http://savannah.gnu.org/bugs/?func=detailitem&item_id=11021), November 2004. Accessed April 2005.
- [2] S. Chazelas. Bugs submitted on <http://savannah.gnu.org/bugs>. Screen Mailing list (gnu.screen.user), <http://article.gmane.org/gmane.comp.gnu.screen.user/456>, January 2005. Accessed April 2005.
- [3] J. R. Daily. screen: Reproducible key sequence causes hard lock. [submit@bugs.debian.org](mailto:submit@bugs.debian.org), <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=157873>, August 2002. Accessed April 2005.
- [4] F. S. Foundation. Gnu screen. <http://savannah.gnu.org/projects/screen/>, April 2003. Accessed April 2005.
- [5] F. S. Foundation. bug #11610 overview: Xon (ctrl-s) halts screen 4.0. [https://savannah.gnu.org/bugs/?func=detailitem&item\\_id=11610](https://savannah.gnu.org/bugs/?func=detailitem&item_id=11610), January 2005. Accessed April 2005.
- [6] Y. Groups. Yahoo! groups: gnu-screen. Via <http://groups.yahoo.com/group/gnu-screen/>, repository via <http://dir.gmane.org/gmane.comp.gnu.screen>, April 2005. Accessed April 2005.
- [7] D. Jacobson. screen: Oliver's version impressions [+jeurgen's]. comp.sources.d, [http://groups.google.ca/groups?selm=1991Aug27Tu192544.Dan\\_Jacobson-%40ATT.COM&output=gplain](http://groups.google.ca/groups?selm=1991Aug27Tu192544.Dan_Jacobson-%40ATT.COM&output=gplain), August 1991. Accessed April 2005.
- [8] O. Laumann. Re: Bugfixes/enhancements to "screen" - (nf). comp.sources.bugs, <http://groups.google.ca/groups?selm=52100003-%40tub.UUCP&output=gplain>, July 1987. Accessed April 2005.
- [9] O. Laumann. Re: enhancements to screen... comp.sources.d, <http://groups.google.ca/groups?selm=701%40tub.UUCP&output=gplain>, October 1988. Accessed April 2005.
- [10] O. Laumann. Re: Patch (really enhancement) for 'screen' v 2.0a (of 19-oct-88). comp.sources.bugs, <http://groups.google.ca/groups?selm=850%40tub.UUCP&output%20=gplain>, May 1989. Accessed April 2005.
- [11] O. Laumann. [c.s.wanted] re: Need latest version of screen. comp.sources.wanted, <http://groups.google.ca/groups?selm=1991Feb8-.032239.1955%40ox.com&output=gplain>, February 1991. Accessed April 2005.
- [12] O. Laumann. Re: Scrollback for screen. alt.sources.d, comp.sources.d, <http://groups.google.ca/groups?selm=4523%40kraftbus.cs.tu-berlin.de&output=gplain>, August 1991. Accessed April 2005.
- [13] O. Laumann. Re: Questions about the birth of screen. Private email between Abram Hindle and Oliver Laumann, March 2005. Accessed March 2005.
- [14] E. S. Raymond. The cathedral and the bazaar. <http://www.catb.org/esr/writings/cathedral-bazaar/>, November 1997. Accessed April 2005.
- [15] M. Schroeder. Re: (no subject). Screen Mailing list (gnu.screen.user), <http://article.gmane.org/gmane.comp.gnu.screen.user/418>, January 2005. Accessed April 2005.
- [16] M. Schroeder. Re: Screen history + questions. Screen Mailing list (gnu.screen.user), <http://article.gmane.org/gmane.comp.gnu.screen.user/477>, February 2005. Accessed April 2005.
- [17] E. Smith. screen refusing to reattach. Screen Mailing list (gnu.screen.user), <http://article.gmane.org/gmane.comp.gnu.screen.user/516>, March 2005. Accessed April 2005.
- [18] J. Weigert. iscreen 3.2. <http://www.mit.edu/afs/net/user/tytso/src/screen-3.2/poster>, May 1992. Accessed April 2005.
- [19] J. Weigert. iscreen 3.2: Changes. <http://www.mit.edu/afs/net/user/tytso/src/screen-3.2/CHANGES>, May 1992. Accessed April 2005.
- [20] J. Weigert and M. Schroeder. GNU Screen. <http://www.gnu.org/software/screen/>, May 2004. Accessed April 2004.

- [21] J. Weigert, M. Schroeder, O. Laumann, and W. Davison. SCREEN(1): screen - screen manager with VT100/ANSI terminal emulation. Screen man page via <http://www.gnu.org/software/screen/>, May 2004. Accessed April 2004 on version 4.0.2.