Accidentally Accessible: a Mostly-FOSS Workflow

Chris Brannon & Sarah Newman

prgmr.com

March 9th 2019

Agenda

- Why care now?
- Introductions
- Communication tools
- Physical infrastructure
- Hiring Considerations

Agenda (continued)

- User interfaces for the blind
- Applications
- Pain Points
- Best practices for screen readers
- Other disabilities

Why Care Now?

- Things you do for accessibility are often reusable
- Opening up your hiring or contributor pool is critical
- Inaccessibility is technical and social debt

About Sarah

- Grew up interested in computers
- Got a degree in computer engineering
- Nokia n800 – debian – got me back into Linux
- Joined prgmr to improve automation

About Chris

- Interested in Linux after working with Unix shell accounts
- Drawn to Linux because it was free and modifiable
- Screen readers for proprietary operating systems were costly ($1000 initially with a $200 per year yearly maintenance agreement)
- Linux was one of the few operating systems that could be easily installed without sight

About prgmr.com

- Provides Virtual Private Servers to the general public
- Founded in 2005 by Luke Crawford
- Started with FreeBSD Jails
- Moved to Xen within a year

About prgmr.com (continued)

- Own all our own equipment
- Operate our own network
- Self-host almost all our own infrastructure
- Rent racks

Communications

- Email – Postfix + Dovecot
- Chat (IRC) – ZNC + InspIRCd
- Voice – traditional phone or Signal
- Documents – plain text, Markdown, or MediaWiki
- Ticketing – Request Tracker

Physical Infrastructure

- Serial consoles for all equipment
- Command line utilities
- Built-in web user interface
- Server installs via kickstart or serial console

Hiring Considerations

- No concerns about day to day operations
- With remote work, data center doesn't matter
- Graphs are a problem
- Really large log files seem challenging

User interfaces for the blind

- Text to speech - 100 to 900 wpm
- Text to braille - 40 to 80 wpm
- Haptic Reading/Feedback - 20 wpm if it existed
- Non-spoken audio feedback (sound icons)

Term Definitions

- Screen readers
  - "Screen reader" is often a misnomer
- Self-voicing applications

Self Voicing vs. Screen Reader Demo

Python “Hello World” in...

- emacs
- nano

Communication Clients

- email – gnus, built in to emacs
- chat – usually erc, also built into emacs. irssi also works
- voice – plain phone or signal
- smart phone – built in screen reader

Communication Clients (continued)

- documents – emacs and MediaWiki via Lynx
- ticketing
  - Custom github client via API
  - Request Tracker – command line client and email, sometimes web

Pain Points

- Log analysis
- Intra-line diff
- Screenshots – OCR doesn't work

Pain Points (continued)

- BIOS
- Graphs

Best Practices for Screen Readers

- Text is always accessible
- Alt-text or labels for user interface elements
- Standard GUI toolkits
  - GTK (Linux)
  - QT5 (Linux, Windows)
- Don't use electron

Web Accessibility

- Web Accessibility Initiative – Accessible Rich Internet Applications
- Keyboard accessibility is the first 90%
- Don't use standard HTML widgets reimplemented in JavaScript

Other Disabilities

- Dyslexia
  - Screen readers might also be used for dyslexia
- Single handed input
- Keyboardless input
- Color blindness
- Low vision

Summary

- Now is a great time to start thinking about accessibility
- All standard communication tools are already accessible
- Programming, include Python, is doable with screen readers
- Affordances for other disabilities are available

Special Thanks to:

- Alison Chaiken
- Kate Deibel, Inclusion & Accessibility Librarian, Syracuse University Libraries
- FOSS Accessibility Maintainers

Closing

Any questions?

Braille displays

- Orbit reader 20, $450
  https://www.aph.org/orbit-reader-20/

Dyslexia

https://www.tandfonline.com/doi/full/10.11120/ital.200