



OPEN FIRMWARE
CONFERENCE
SOURCE
2020 DECEMBER
01 - 03

Webboot



SPEAKER

Shiyu Xu

Gabriel Biren

Overview

- *Webboot* is an open source source tool to boot live OS distros from the web.
 - Bootloader that boots directly from ISOs
 - Able to connect to internet and download new ISOs
- Part of *u-root*, meant to be used with *LinuxBoot*
 - Written in Go
 - Uses packages from *u-root*
- Open Source
 - Available: github.com/u-root/webboot



Use Cases

- Multi-boot USB stick
 - Quickly switch between OS's
 - Ex. Boot Ubuntu, power off, boot Arch
- Loaner laptop (esp. with sensitive work)
 - Power on, user selects an OS and works
 - Power off, file system is wiped



Advantages

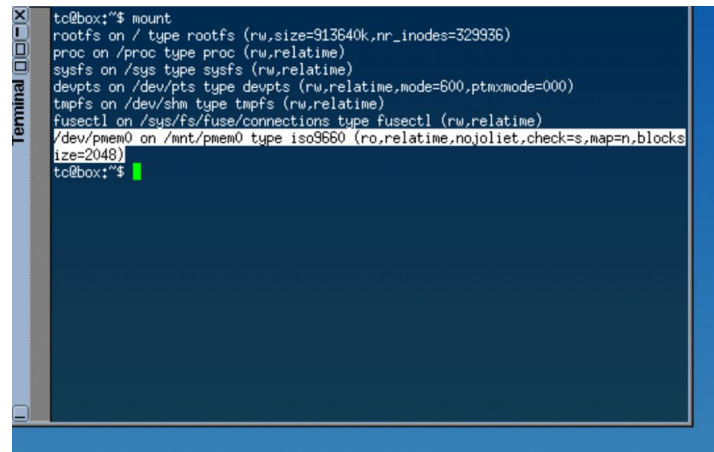
- Text-based UI
 - User friendly
 - No need to remember command line parameters
 - Easy to read messages and alerts
- Simple OS Management
 - Installing/removing OS usually involves partitioning and formatting the boot device
 - Add a new OS by copying the ISO
 - Remove an existing OS by removing the ISO
- Statelessness
 - No installation
 - Temporary file system in memory

Original Concept

1. Download a Linux ISO.
2. Copy the ISO to a persistent memory device.
 - a. Allocated region of memory that is treated like a block device.
3. Retrieve the ISO's kernel and initrd.
4. Use *kexec* to boot into the new distribution.
5. New distribution initializes.
 - a. Locates ISO in the persistent memory device.

Results

- Presented *Webboot* at OSFC 2019.
- Able to boot into a modified Tinycore ISO.
 - Recompiled the kernel with persistent memory support.

A terminal window titled 'Terminal' with a dark blue background and white text. The terminal shows the output of a 'mount' command. The output lists several filesystems being mounted: rootfs on /, proc on /proc, sysfs on /sys, devpts on /dev/pts, tmpfs on /dev/shm, fusectl on /sys/fs/fuse/connections, and /dev/pmem0 on /mnt/pmem0. The last line shows the prompt 'tc@box:~\$' with a green cursor.

```
tc@box:~$ mount
rootfs on / type rootfs (rw,size=913640k,nr_inodes=329936)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600,ptmxmode=000)
tmpfs on /dev/shm type tmpfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
/dev/pmem0 on /mnt/pmem0 type iso9660 (ro,relatime,nojoliet,check=s,map=n,block
size=2048)
tc@box:~$
```



Issues

- Most distributions don't support persistent memory by default.
 - Persistent memory device is not recognized when the distro boots.
 - Lose the ISO and distribution goes to recovery shell.
- Persistent memory is difficult to work with.
 - Need to know size ahead of time.
- Need to redownload the ISO every time.
- Command-line program is not user friendly.
 - Need to know how to use command line, what arguments to use, etc.



What we did this year.



UI

- Based on *TermUI*
- 3 basic UI components
 - Menu
 - Input box
 - Message box
- Features
 - Multi-hierarchy menu
 - Return to the previous menu screen
 - Support mouse scroll or PgUp/PgDn

UI Components

- *Menu*: title and list of menu options
 - Select an option by entering the corresponding number
 - Scroll with mouse or PgUp/PgDn
- *Input Box*: retrieve text input
 - Can create with *isValid()* function to validate the input
 - Display warning if input is not valid
- *Message Box*: display long messages and alerts
 - Supports scrolling

```
Webboot--0/2
[0] Use Cached ISO
[1] Download an ISO

Choose an option:


<Esc> to go back, <Ctrl+d> to exit
```

```
Enter password:


<Esc> to go back, <Ctrl+d> to exit
```

```
Message--0/3
mount /tmp/mnt-704958718: from device "/dev/loop4" (fs type iso9660, flags 0x401): invalid argument
```

Download Progress

- Special type of message box
- Process
 - Get the total size of the item to download
 - Get size of each piece of the data stream as it downloads
 - Calculate the download process as a percentage
 - Dynamically update the content of message box

```
-Operation Running-
Downloading... 1.82% (12.992 MB)

Press <Esc> to cancel.
```

UI Test

- Goal: Simulate user input
 - Simulating keyboard and mouse interrupt events is too complicated
- Solution
 - Use TermUI to draw the UI
 - TermUI will parse the user input from a channel
 - Replace the original channel with our own test channel
- Do not directly use TermUI, but manually call TermUI's function to parse the test input channel
 - Manually implement the processes we want to test
 - E.g. {'0', '<Enter>', '1', '1', '<Enter>'}

UI Test - Case Study

- Create a numbered menu by passing an array *entries*
 - *Entries* array can be any size
 - Menu displays 10 options at a time
 - User can scroll if *entries* has more than 10 options
 - Keep track of *first* and *last* indexes to display
- When user types their selection, perform basic validation
 - Ex. If there are 5 options, user cannot select Option 22

UI Test - Case Study

- Tried to add a new option to a menu
 - UI test caught an error
- Issue: Unit test tries to select the last option
 - Only 10 options displayed on screen
 - 11 options were now available
 - `if err == nil && c >= first && c < last { // valid option }`
 - First = 0
 - Last = 10
 - C = 10
- UI logic is confusing: existing option cannot be selected.



UI Test - Case Study

- ***UI test caught a UI logic problem before users did***
- Reconsider UI decisions as project develops
 - Logic made sense when we only had 2 distros to choose from
 - Didn't work when we had 11 options
 - Fix: Any option can be selected now, even if it's not currently on screen

WiFi Setting

- Previously needed to set up WiFi connection before starting *Webboot*.
 - Command line *wifi* utility
- Integrated into our menu flow.
 - Scan for networks.
 - Select ESSID.
 - Collect credentials.
 - Different for NoEnc, PSK, EAP.
 - Connect to WiFi.

```
—Wireless Networks---0/13—
[0] BIREN1: WPA-PSK (only passphrase)

[1] MySpectrumWiFi88-2G: WPA-PSK (only passphra...
[2] DIRECT-71-HP OfficeJet 3830: WPA-PSK (only ...
[3] MySpectrumWiFi10-2G: WPA-PSK (only passphra...
[4] ORBI55: WPA-PSK (only passphrase)

[5] ATTUPuBwui: WPA-PSK (only passphrase)

[6] MySpectrumWiFi88-5G: WPA-PSK (only passphra...

Choose an option

<Esc> to go back, <Ctrl+d> to exit
```




Local Cache

- Avoid redownloading ISO every time.
 - Ubuntu is ~2.5G
 - At 2MB/s, ~20 min download
- Create a /Images directory on the boot device.
- User has two options
 - Download a new ISO and save to cache
 - Retrieve a downloaded ISO from the cache

Alternative to Persistent Memory

- Avoid using persistent memory.
 - Issue: Persistent memory device acted like an emulated CD.
 - Need a way to direct distribution to its ISO.
- *iso-scan/filename=* kernel parameter
 - Available to Ubuntu-based distributions
 - Provide a path to ISO
 - Mount all devices and search for path
 - Ex. `iso-scan/filename=/Images/Ubuntu.iso`

Kernel Parameters

| Distribution Family | Kernel Parameter(s) |
|---------------------|--|
| Arch | img_dev=/dev/disk/by-uuid/UUID img_loop=PATH_TO_ISO |
| CentOS | iso-scan/filename=PATH_TO_ISO |
| Debian | findiso=PATH_TO_ISO |
| Tinycore | iso=UUID/PATH_TO_ISO |
| Ubuntu | iso-scan/filename=PATH_TO_ISO |

Boot Process

1. Select a Linux ISO.
 - a. Download or retrieve from local cache.
2. Mount the ISO.
3. Parse the GRUB/SYSLINUX config file.
 - a. Locate the kernel, initrd, and kernel parameters.
4. Determine the *iso-scan=* kernel parameter.
 - a. Append to the kernel parameters retrieved in Step 3.
5. Use *kexec* to boot into the distribution.



Roadmap

- Support additional distributions.
 - Currently support Arch, CentOS, Debian, Tinycore, Ubuntu.
- Verify ISO checksums.
- Include *Webboot* in the *U-root* init process.
 - Power on computer, go straight to *Webboot* menu.
- Always looking for new contributors.
 - Repo: github.com/u-root/webboot



Thank You!
(esp. to Ron Minnich, Prachi Laud, NERF Team)