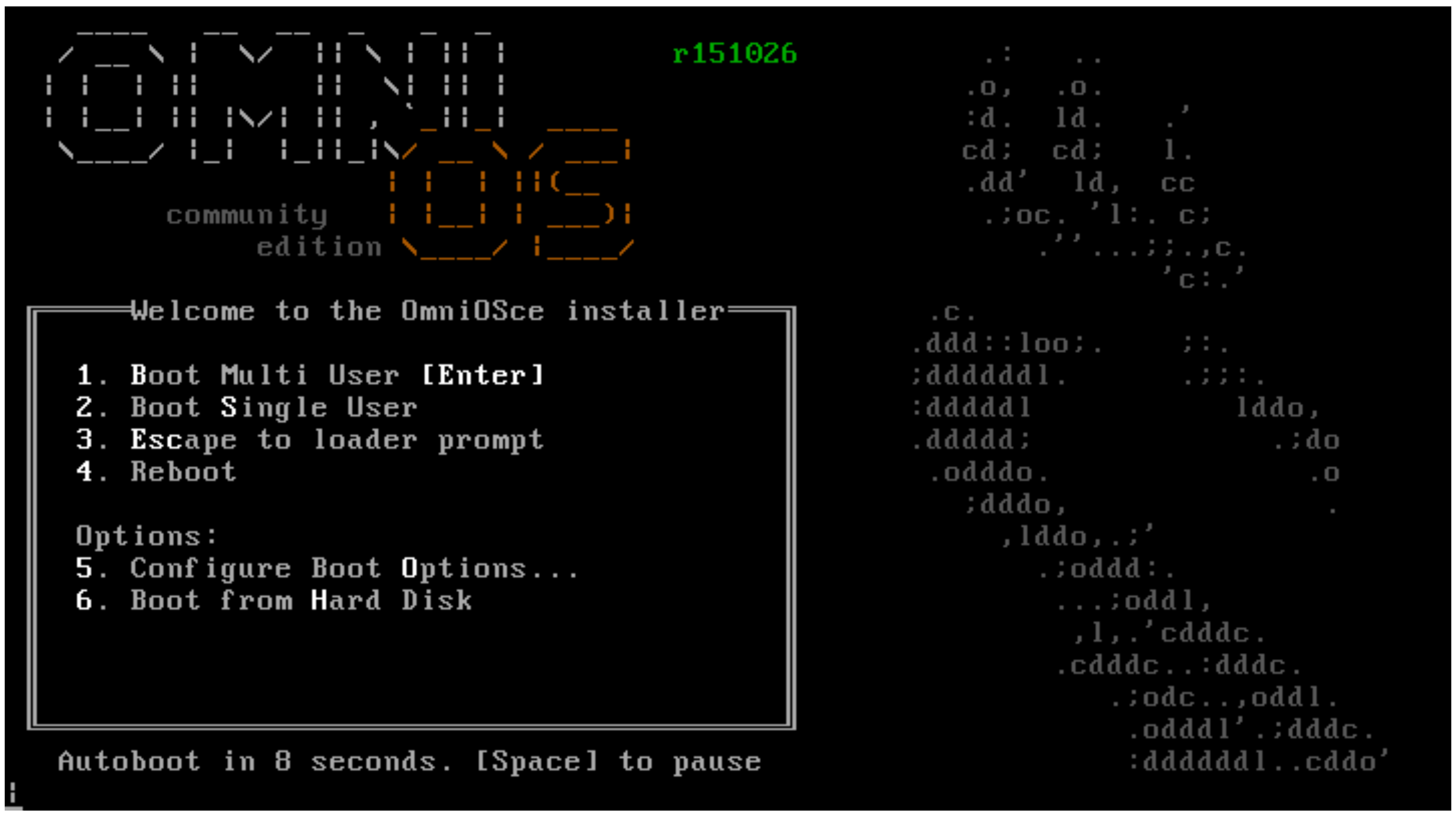
Is There a New Loader in FreeBSD 13.0?

BY TOOMAS SOOME

The short answer is “no, it is still the same good old loader.” But we are trying to make it more friendly and support more features.

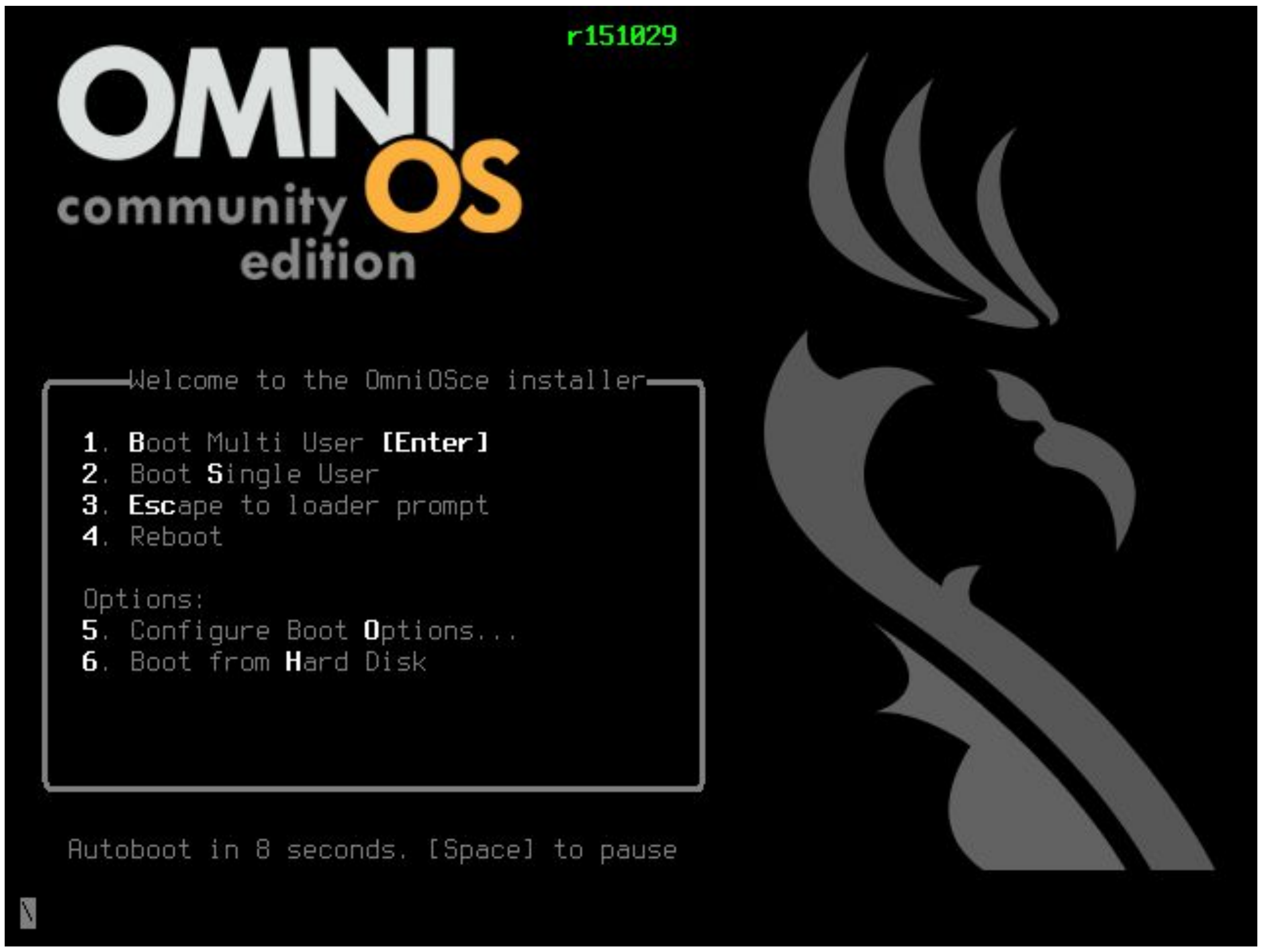
I started working with boot loaders that were not on FreeBSD but on illumos. At that time, illumos was using old grub 0.96 and supported only the BIOS boot. UEFI systems were already around at that time and illumos really needed to support UEFI systems. My initial work was to investigate newer grub. It was feature-rich, widely used but hard to manage, and its licensing is not friendly when you need to support features specific to file systems like zfs or to add operating-system specific features. This led me to the FreeBSD boot loader and contributing to its development.

When I started porting the FreeBSD boot loader to illumos, illumos supported a serial console and VGA text mode console only. To support UEFI, I had to implement support for the UEFI framebuffer-based console for the illumos kernel. Once implemented, a logical step was to add the same feature for the loader, and when there is an option to draw the console on the UEFI framebuffer, then re-using the same code to draw on the Vesa BIOS Extensions (VBE) linear framebuffer is just another logical step in development. Once done, we got public postings like this https://omnios.org/setup/fb:



```
                                      r151026

        community
            edition

  Welcome to the OmniOSce installer

    1. Boot Multi User [Enter]
    2. Boot Single User
    3. Escape to loader prompt
    4. Reboot

    Options:
    5. Configure Boot Options...
    6. Boot from Hard Disk

  Autoboot in 8 seconds. [Space] to pause
```

Figure 1 Loader with ascii art.



```
Welcome to the OmniOSce installer
 1. Boot Multi User [Enter]
 2. Boot Single User
 3. Escape to loader prompt
 4. Reboot

 Options:
 5. Configure Boot Options...
 6. Boot from Hard Disk

Autoboot in 8 seconds. [Space] to pause
```

Figure 2 Loader with images.

Back to the FreeBSD Boot Loader

But enough about illumos, let's get back to FreeBSD and see what we have managed to do there. Please note, most of the work I have done has involved flowing from Freebsd to illumos or vice versa.

OpenZFS

As FreeBSD 13.0 is now using OpenZFS, we support most OpenZFS features for booting. The encrypted datasets and draid are still in the todo list, however.

Console

Literally, the most visible change is the graphical console for the loader. While the current implementation is not perfect and can be improved, I hope most users will enjoy the updated look.

The loader console terminal emulator is teken from the kernel tree. From there, we have the first two tunables we can set:

```
teken.fg_color
teken.bg_color
```

The acceptable values are ansi color names or numeric values 0 – 7.

The UEFI loader uses the framebuffer console by default unless the serial console is configured.

The BIOS loader defaults to use text mode at this time. For the BIOS loader, the console can be controlled by setting:

```
screen.textmode="0"
```

This will cause the loader to set up VBE framebuffer mode and to use a display-preferred resolution when EDID information is available. The default fall back resolution is 800x600.

Both UEFI and BIOS loaders allow setting the screen resolution via the following tunables:

```
efi_max_resolution
vbe_max_resolution
```

Having vbe_max_resolution set will also cause the loader to switch to use VBE framebuffer mode.

The framebuffer mode can also be set, queried and support modes listed by platform specific commands.

Command `gop` allows a user to get, set and list resolutions in the UEFI loader. Command `gop off` will switch the loader from drawing the console to the UEFI built-in terminal output method—Simple Text Output Protocol.

Command `vbe` allows the user to get, set and list resolutions in the BIOS loader. Command `vbe on` will switch the loader to use VBE framebuffer and `vbe off` will switch the loader to use VGA text mode.

If a user has switched the BIOS loader to use the VBE framebuffer but is booting an older kernel that does not provide a VT vbefb driver, then the loader will switch the console to VGA text mode just before starting the loaded kernel.

Fonts

At this time, we are providing terminus family console fonts, installed into the /boot/fonts directory. The loader has a built-in 8x16 font, but to save space, the built-in font only provides an ASCII set.

After the loader starts and initializes and it has obtained access to disks and determined the boot device and a boot file system, the loader will search for /boot/fonts directory and INDEX.fonts file. If present, the loader will get the list of available fonts and will build an internal, indexed list of available fonts. The INDEX.fonts file is used because we need to support tftp file transport, but tftp protocol does not implement reading directory listings.

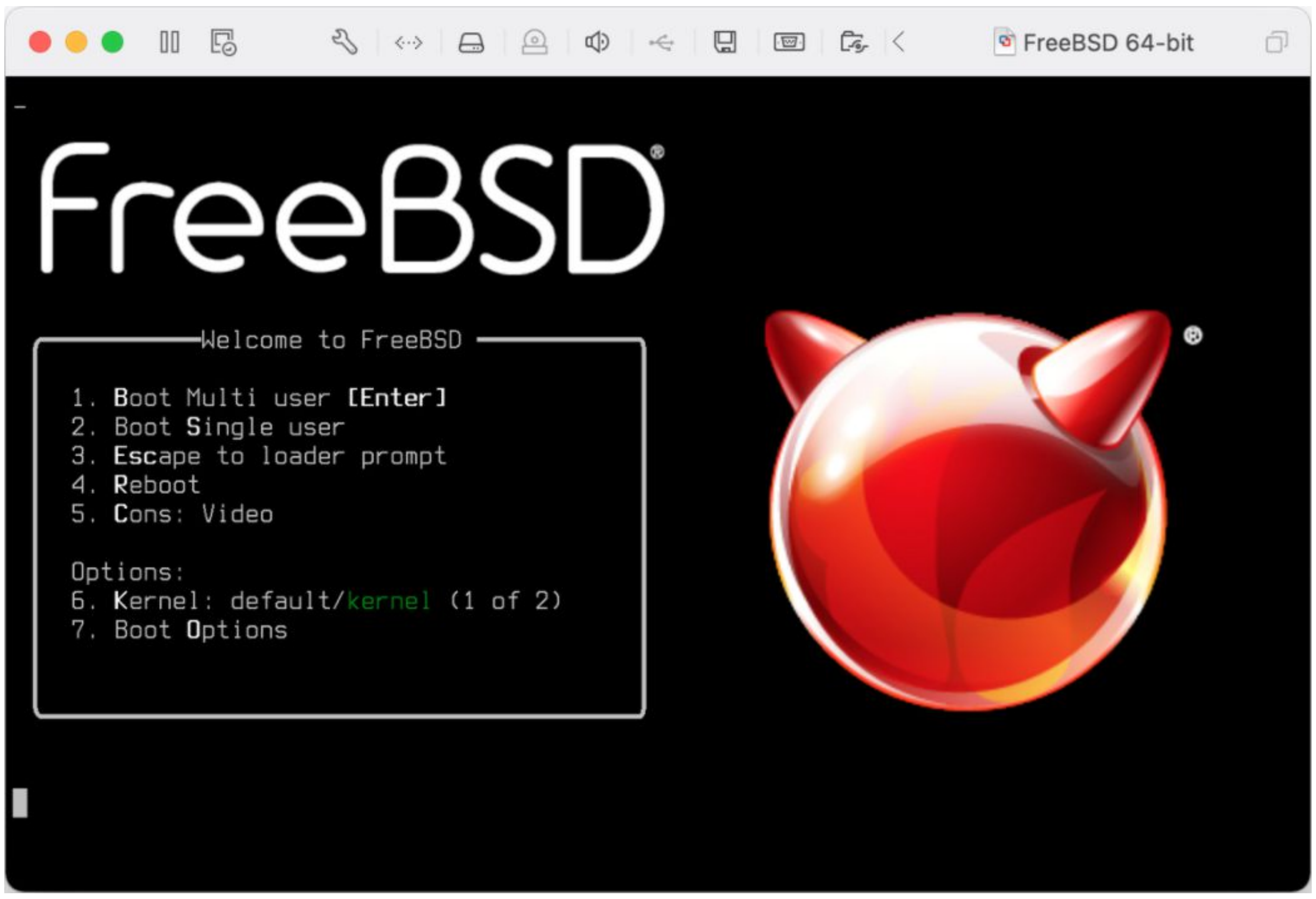
Once we have a list of available fonts, we load a preferred font based on the resolution used on the console display. If the default selection is not working well for a user, there are two methods for changing defaults:

First, the environment variable screen.font appears and permits changing a used font. An attempt to use a bad value or to unset the value will cause the list of currently available fonts to be printed on the console.

Second, the command `loadfont` allows a user to load a custom font file such as `/boot/fonts/gallant.fnt`. The font needs to be prepared with the `vtfontcvt(8)` tool.

The font indexing in the loader assumes unique font sizes. When there is an already-registered font file for font size 8x16, an attempt to load a different font file providing the same font size will cause the previously loaded file to be replaced by a new file.

The font currently used by the loader will be passed to the loaded kernel. By doing so, we preserve the look and feel and achieve a consistent transition from the boot loader to the running operating system.



```
Welcome to FreeBSD

1. Boot Multi user [Enter]
2. Boot Single user
3. Escape to loader prompt
4. Reboot
5. Cons: Video

Options:
6. Kernel: default/kernel (1 of 2)
7. Boot Options
```

Figure 3 FreeBSD 13.0 Boot loader

Images

To build better looking screens, the loader supports display of PNG-formatted image files. At this time, we require TrueColor with an alpha channel, and we support image scaling. Examples of how to use images in a logo or brand components of the bootloader menu can be found from drawer.lua and gfx-orb.lua files.

Drawbacks

Some systems experience a slow console when the framebuffer console is used. With the VBE framebuffer, one possible workaround is to use smaller color depth—default is 32-bit colors. With both VBE and UEFI, it may be possible to use smaller resolution and to configure better resolution once the KMS driver is running. And, of course, in some cases the only reasonable option may be to use a text console.

Summary

So again, no, there is not a new boot loader in FreeBSD, but we are trying to make sure it does what it should do—which is support loading and booting the FreeBSD operating system, provide features people might need for the task, and look reasonably good.

TOOMAS SOOME Born in Estonia. Toomas has been a UNIX admin since 1993, working mainly with Solaris. He is also an infrastructure architect, illumos developer and a FreeBSD src committer. Toomas is not afraid of boot loaders and can read and write Forth.