UBUNTU - BOOT TO RAM - COUPLE OF NOTES

This post contains couple of notes on this topic in form of very short guide.

Requirements

- VirtualBox
- Ubuntu Minimal CD

First step - create virtual machine

Create new virtual machine using VirtualBox and install minimal Ubuntu OS.

Second step - install required packages

Install *squashfs-tools* package so you can create squashfs image later:

```
$ sudo apt-get install squashfs-tools
```

Install *live-boot* package with dependencies (*live-boot-initramfs-tools*) so you can use boot to ram:

```
$ sudo apt-get install live-boot
```

Third step - prepare image contents

Create new directory and copy root file system contents:

```
$ sudo mkdir /squashfs
```

You need to exclude contents of directory */live* and one created just moment ago.

You can also exclude contents of directories like */boot/**, */tmp/**, ...

```
$ sudo rsync -a --delete --one-file-system / /squashfs \

    --exclude=/live --exclude=/squashfs
```

It's good idea to remove root file system from */squashfs/etc/fstab* file now.

Fourth step - create squashfs image

Create */live* directory (squashfs image will be stored here):

```
$ sudo mkdir /live
```

Create squashfs image:

```
$ sudo mksquashfs /squashfs /live/livefs.squashfs -noappend -always-use-fragments
```

Fifth step - configure grub2

Change *GRUB_TIMEOUT* to *-1* in */etc/defaults/grub* so it will be waiting forever in boot menu.

Update grub configuration:

```
$ sudo update-grub
```

Check your kernel release:

```
$ uname -r

3.0.0-17-generic
```

Edit */etc/grub.d/40_custom* file to add new entry in grub2 menu and take into account your kernel release:

```
menuentry "Live minimal OS" {
 set root='(hd0,1)'
 linux /boot/vmlinuz-3.2.0-24-generic boot=live toram=livefs.squashfs
 initrd /boot/initrd.img-3.2.0-24-generic
}
```

Update grub configuration again:

```
$ sudo update-grub
```

Sixth step - check it out!

Reboot system and check it out:

```
$ sudo reboot
```

Couple of notes

Compression is very effective as it can compress 2 GB file system (KDE + Libre Office + couple of smaller applications) to around 700 MB.

System becomes blazingly fast and you can easily create live usb this way.

To make your changes persistent just create partitions labelled accordingly *live-rw* for root, *home-rw* for home file system and add *persistent* parameter to kernel boot parameters. If you can't create new partitions then create files in root directory (not in live fs). For example to create persistent home (200MB, without reserved blocks) use commands:

```
$ sudo dd if=/dev/zero of=/home-rw bs=1M count=200

$ sudo mkfs.ext4 /home-rw

$ sudo tune2fs -m 0 /home-rw

$ sudo tune2fs -L home-rw /home-rw
```

Apparmor doesn't work very well so you need to remove it.

To see how it works go to directory */usr/share/initramfs-tools/scripts* and start with reading *live* file. After any modifications don't forget to update initramfs:

```
$ sudo update-initramfs -u
```

Don't forget to read *live-boot* manual page.

Couple of common errors

```
only one RO file system supported with exposedroot
```

Just remove *livefs.squashfs* file from */squashfs/live/* directory and create squashfs image again.

```
a wrong rootfs was mounted
```

This error means that *live-boot* package was not installed on system that was used to create squashfs image. Just install it and create image again.

Source: https://blog.sleeplessbeastie.eu/2012/05/02/ubuntu-boot-to-ram-couple-of-notes/