Loopback Filesystems (Lab)

coursera.org/learn/linux-tools-for-developers/supplement/56Txd/loopback-filesystems-lab

Exercise

Linux systems often use **loopback** filesystems, in which a normal file is treated as an entire filesystem image.

First, create an empty file by doing:

```
$ dd if=/dev/zero of=/tmp/part count=500 bs=1M
```

which will create an empty 500 MB file named **/tmp/part**. You can adjust the size if you are short on space.

You can then put an **ext4** filesystem on the file by doing:

```
$ mkfs.ext /tmp/part
```

which you can then mount by doing:

```
$ mkdir /tmp/mntpart
$ sudo mount -o loop /tmp/part /tmp/mntpart
$ df
Filesystem    Type   1K-blocks      Used Available Use% Mounted on
/dev/sda5     ext3    10157148   6238904   3393960  65% /
....
/tmp/part     ext3      495844     10544    459700   3% /tmp/mntpart
```

Once it is mounted, you can create files on it, etc., and they will be preserved across remount cycles.

You can check the filesystem by doing:

```
$ sudo umount /tmp/mntpart
$ fsck.ext4 -f /tmp/part
```

and get additional information by doing:

```
$ dumpe2fs /tmp/part
```

and change filesystem parameters by doing:

```
$ tune2fs /tmp/part
```

For example, you could change the **maximum-mount-count** or **reserved-blocks-count** parameters.

Loopback filesystems have lower performance due to having to pass through the actual filesystem, but are still quite useful.