

Si, por alguna razón macabra, queremos llenar de ceros todos los chips de almacenamiento integrados en el router BananaPi BPI-R3, seguimos estos pasos:

Quemamos una imagen de OpenWrt en la MicroSD.

Ponemos todos los jumpers arriba para iniciar desde la MicroSD e iniciamos el router.

Nos conectamos mediante SSH como root.

Listamos la cantidad de bloques disponibles para borrar, ejecutando:

```
ls -lh /dev/mtd*
```

La salida es la siguiente:

```
/dev/mtd0
/dev/mtd0ro
/dev/mtd1
/dev/mtd1ro
/dev/mtd2
/dev/mtd2ro
/dev/mtd3
/dev/mtd3ro
/dev/mtdblock0
/dev/mtdblock1
/dev/mtdblock2
/dev/mtdblock3
```

Tenemos entonces 4 dispositivos de bloque que borrar, 1 por cada partición. Pero nos llama la atención que, por cada MTD (Memory Technology Device) disponible, hay un archivo «ro». Esto nos da la pauta de que las particiones podrían estar protegidas contra escritura, por lo que más adelante deberemos «desprotegerlas». De todos modos, para saber si las particiones son «escribibles» o no, podemos ejecutar:

```
for vMTD in /dev/mtd[0-9]; do mtdinfo $vMTD; done
```

La salida será esta:

```
mtd0
Name:                ubi
Type:                nand
Eraseblock size:    131072 bytes, 128.0 KiB
Amount of eraseblocks: 980 (128450560 bytes, 122.5 MiB)
Minimum input/output unit size: 2048 bytes
Sub-page size:      2048 bytes
OOB size:           64 bytes
Character device major/minor: 90:0
Bad blocks are allowed: true
Device is writable: true

mtd1
Name:                fip
Type:                nand
Eraseblock size:    131072 bytes, 128.0 KiB
Amount of eraseblocks: 16 (2097152 bytes, 2.0 MiB)
Minimum input/output unit size: 2048 bytes
```

```

Sub-page size:          2048 bytes
OOB size:              64 bytes
Character device major/minor: 90:2
Bad blocks are allowed: true
Device is writable:    false

mtd2
Name:                  reserved
Type:                  nand
Eraseblock size:      131072 bytes, 128.0 KiB
Amount of eraseblocks: 24 (3145728 bytes, 3.0 MiB)
Minimum input/output unit size: 2048 bytes
Sub-page size:        2048 bytes
OOB size:              64 bytes
Character device major/minor: 90:4
Bad blocks are allowed: true
Device is writable:    true

mtd3
Name:                  bl2
Type:                  nand
Eraseblock size:      131072 bytes, 128.0 KiB
Amount of eraseblocks: 4 (524288 bytes, 512.0 KiB)
Minimum input/output unit size: 2048 bytes
Sub-page size:        2048 bytes
OOB size:              64 bytes
Character device major/minor: 90:6
Bad blocks are allowed: true
Device is writable:    false

```

Donde podemos ver que sólo mtd0 es escribible. Identificamos a qué memoria corresponde cada mtddblock:

```
cat /proc/mtd
```

La salida es la siguiente:

```

dev:  size      erasesize name
mtd0: 07a80000 00020000 "ubi"
mtd1: 00200000 00020000 "fip"
mtd2: 00300000 00020000 "reserved"
mtd3: 00080000 00020000 "bl2"

```

Antes de borrar nos aseguramos de desmontar los 4 bloques asignados a esos 4 dispositivos (si es que están montados):

```

umount /dev/mtdblock0
umount /dev/mtdblock1
umount /dev/mtdblock2

```

```
umount /dev/mtdblock3
```

Ahora si, borramos y rellenamos con ceros cada bloque:

```
chmod +w /sys/block/mtdblock0/ro
echo 0 > /sys/block/mtdblock0/ro
flash_erase /dev/mtd0 0 0
dd if=/dev/zero of=/dev/mtdblock0 # Puede tardar hasta 10 minutos
```

```
chmod +w /sys/block/mtdblock1/ro
echo 0 > /sys/block/mtdblock1/ro
flash_erase /dev/mtd1 0 0
dd if=/dev/zero of=/dev/mtdblock1
```

```
chmod +w /sys/block/mtdblock2/ro
echo 0 > /sys/block/mtdblock2/ro
flash_erase /dev/mtd2 0 0
dd if=/dev/zero of=/dev/mtdblock2
```

```
chmod +w /sys/block/mtdblock3/ro
echo 0 > /sys/block/mtdblock3/ro
flash_erase /dev/mtd3 0 0
dd if=/dev/zero of=/dev/mtdblock3
```