Distributed Systems

Inga Rüb

28 November 2018

Coffee?

A filesystem tailor-made for microcontrollers.

Nicolas Tsiftes, Adam Dunkels, Zhitao He, Thiemo Voigt

Enabling Large-Scale Storage in Sensor Networks with the Coffee File System

IPSN 2009

https://github.com/contiki-os/contiki/wiki/File-systems#Coffee

http://dunkels.com/adam/tsiftes09enabling.pdf

https://anrg.usc.edu/contiki/index.php/Contiki_Coffee_File_System

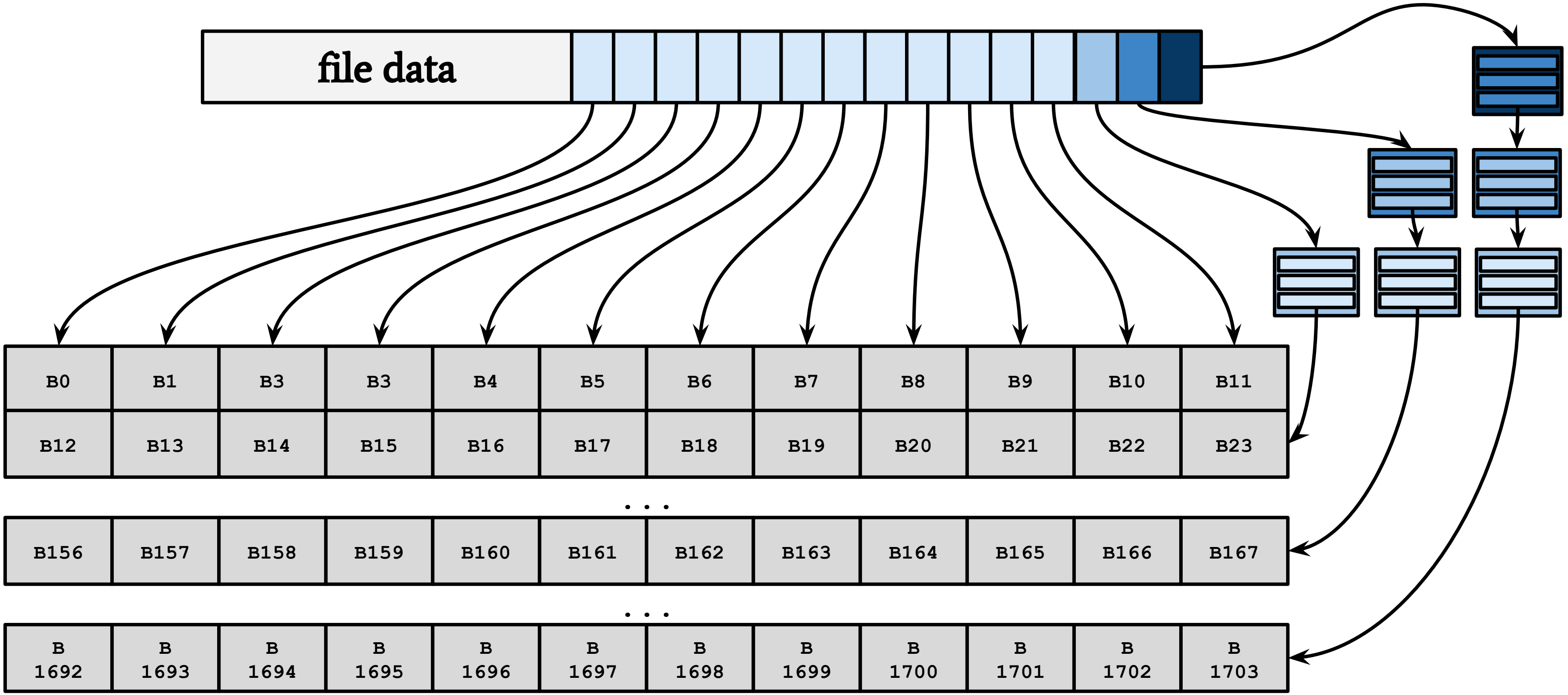
Why we need a filesystem?

- ★ Some sensor networks are storage-oriented rather than communication-oriented.
- ★ We may want to use flash memory as virtual memory backend.

Why we need a **special** filesystem?

1. RAM is limited, hence conventional buffers and inode data is too much.

Conventional File System



B0	B1	B3	B3	B4	B5	B6	B7	B8	B9	B10	B11
B12	B13	B14	B15	B16	B17	B18	B19	B20	B21	B22	B23

. . .

B156	B157	B158	B159	B160	B161	B162	B163	B164	B165	B166	B167

. . .

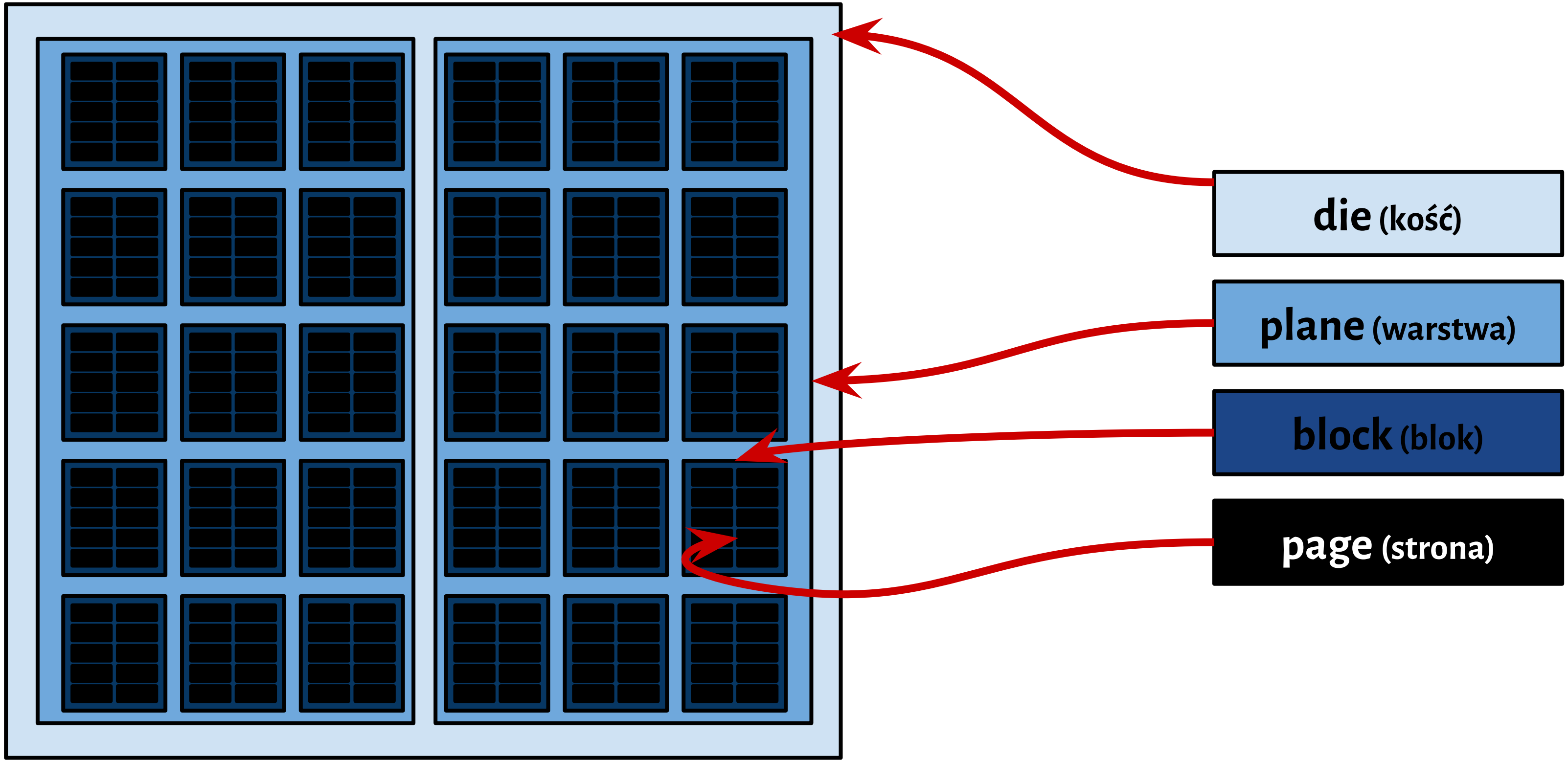
B 1692	B 1693	B 1694	B 1695	B 1696	B 1697	B 1698	B 1699	B 1700	B 1701	B 1702	B 1703

Why we need a **special** filesystem?

1. RAM is limited, hence conventional buffers and inode data is too much.

2. Write operations for flash memory are different from magnetic-disk operations.

Flash is special



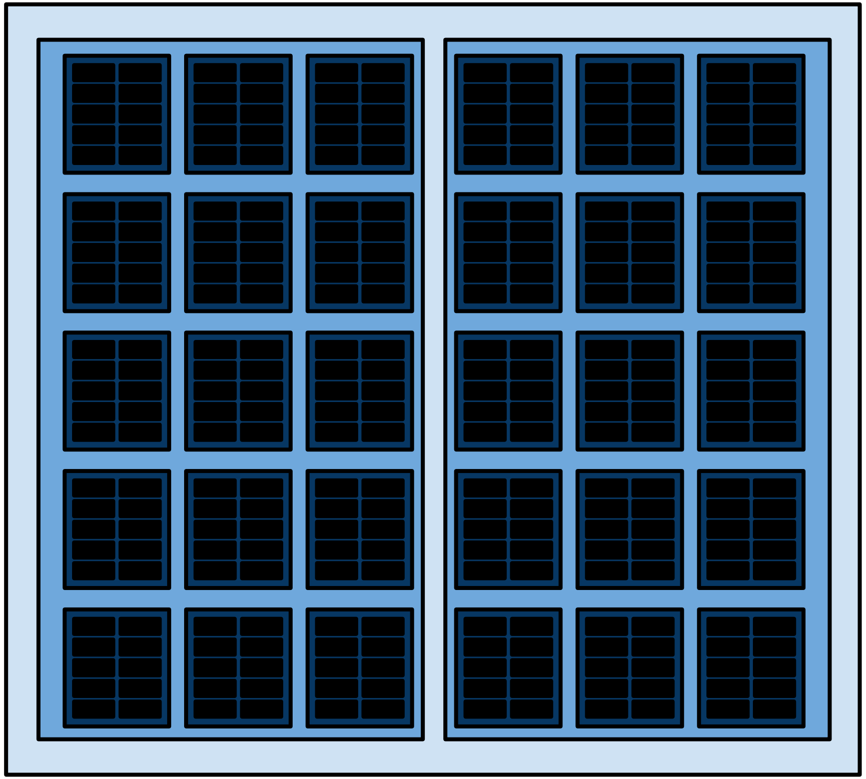
die (kość)

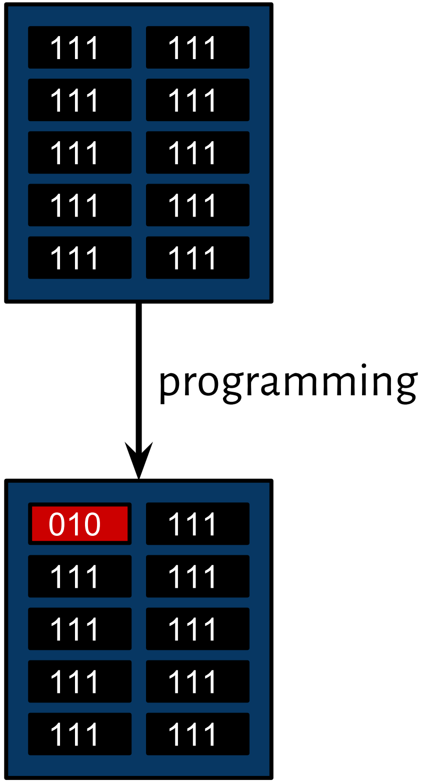
plane (warstwa)

block (blok)

page (strona)

Flash is special



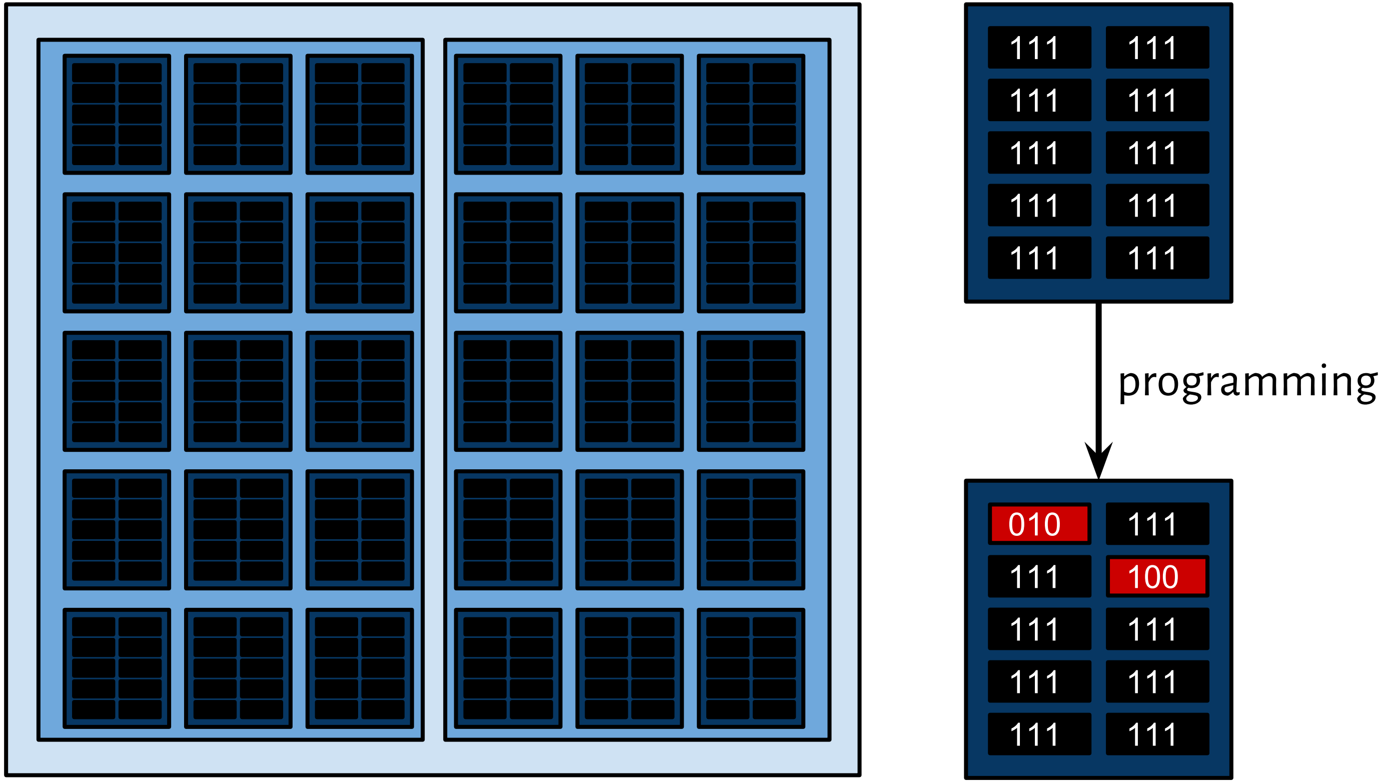


111	111
111	111
111	111
111	111
111	111

programming

010	111
111	111
111	111
111	111
111	111

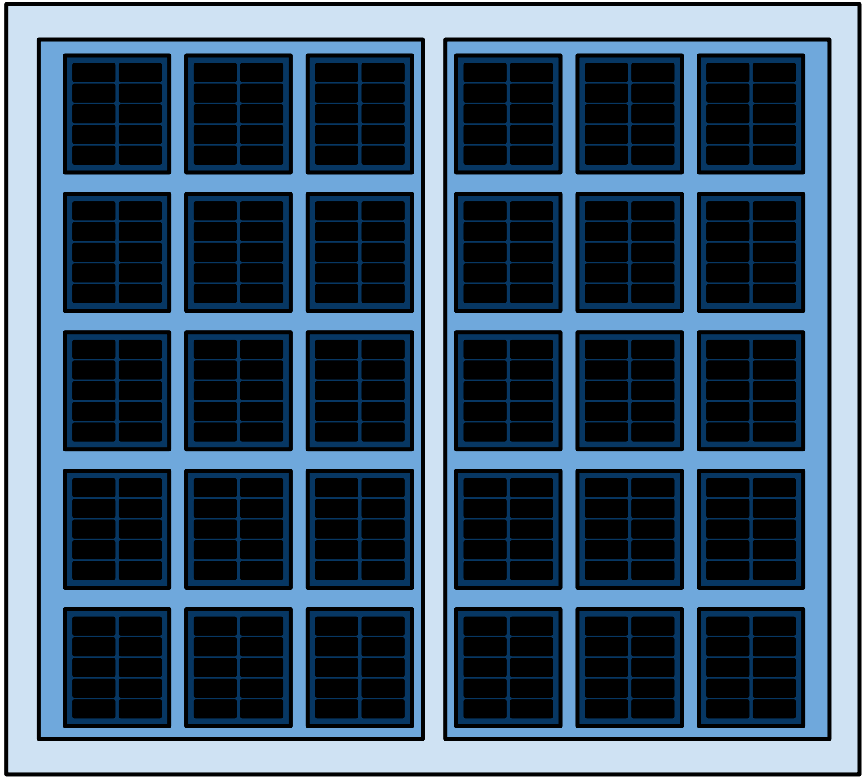
Flash is special

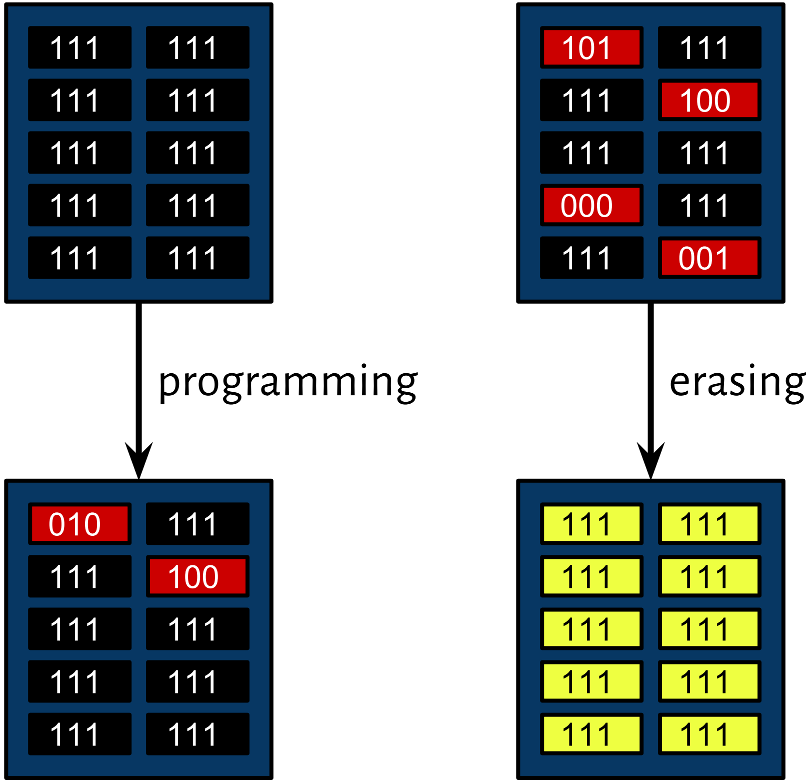


111	111
111	111
111	111
111	111
111	111

010	111
111	100
111	111
111	111
111	111

Flash is special





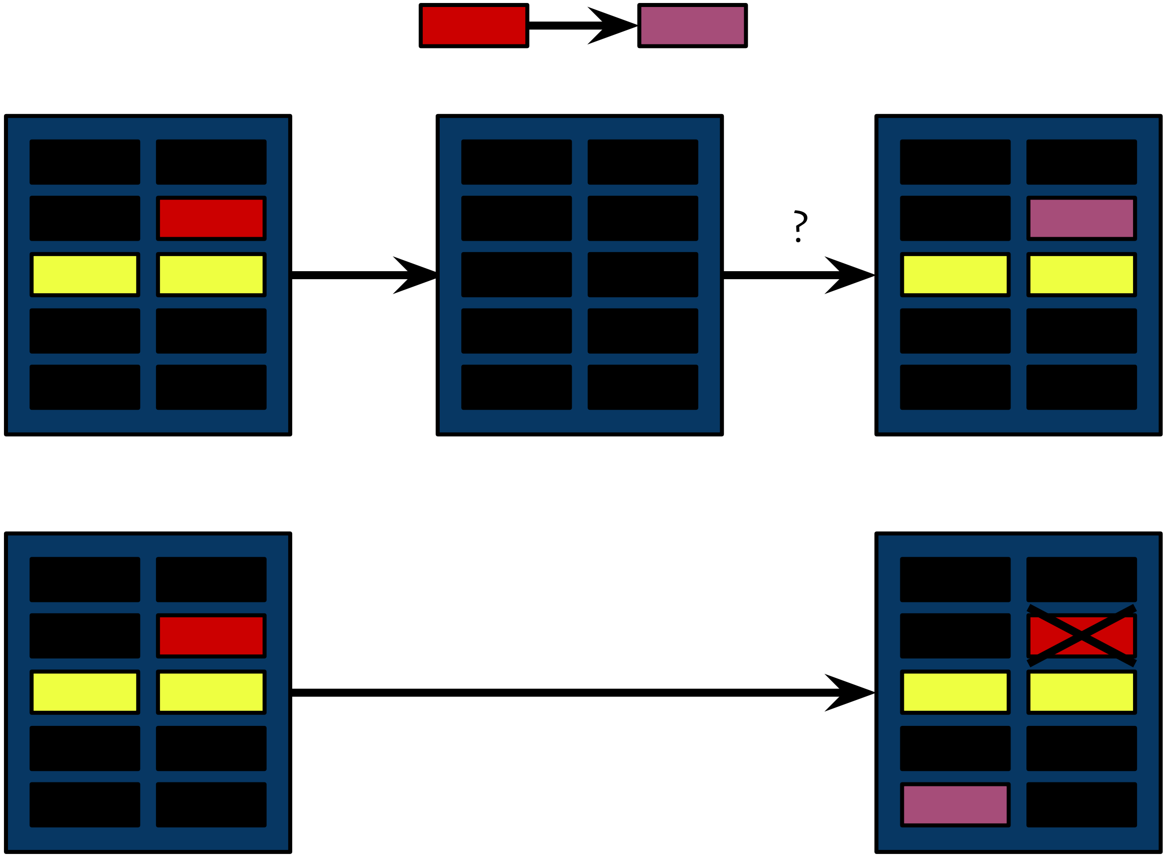
111 111
111 111
111 111
111 111
111 111

010 111
111 100
111 111
111 111
111 111

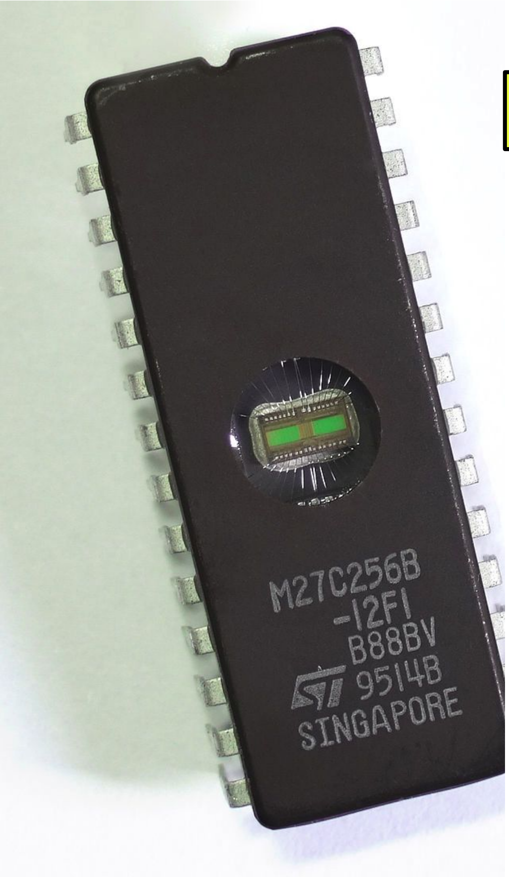
101 111
111 100
111 111
000 111
111 001

111 111
111 111
111 111
111 111
111 111

Reprogramming in flash is special



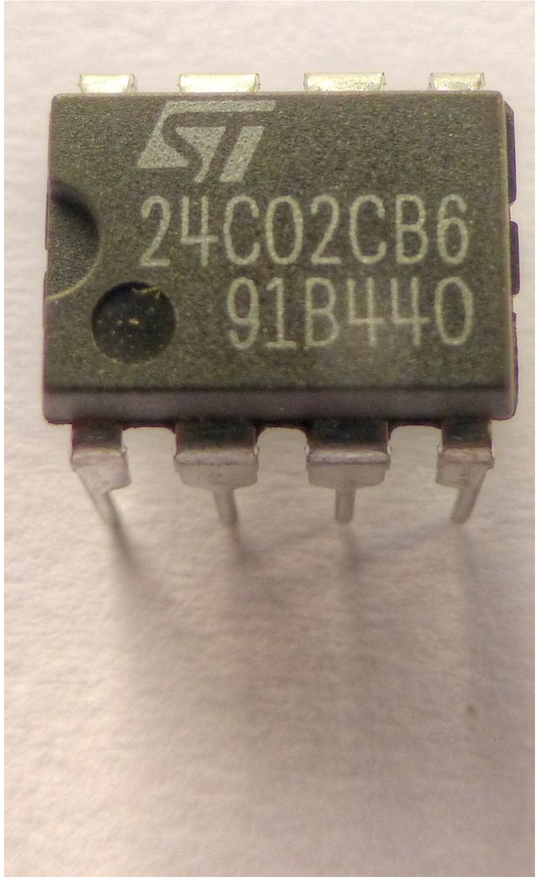
Erasing data - history



EPROM

Erasable
Programmed
Read-Only
Memory

device-wise erasure

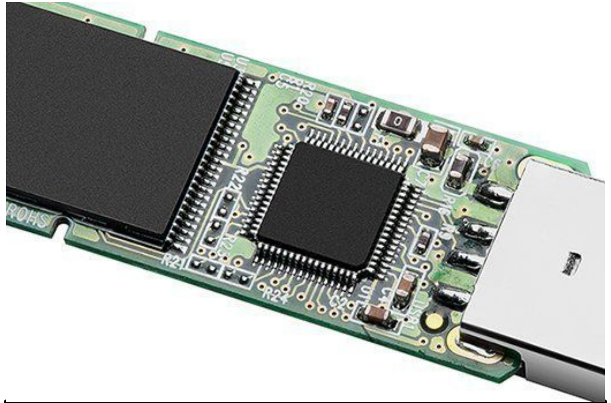


EEPROM

Electrically
Erasable
Programmed
Read-Only
Memory

byte-wise erasure

Erasing data - history



Flash Memory

block-wise erasure

NOR

cells linked in parallel

byte-wise
reading and programming

NAND

cells linked in series

page-wise
reading and programming

Why do we use flash in motes?

- ★ Flash is chip, dense and fast (sometimes).
- ★ Flash memory bits can be programmed with a low overhead.

- ★ Resetting programmed bits requires an expensive sector erase.
- ★ The erase takes time and causes sectors to wear out after erasing many times.
- ★ Fine-grained modifications are more complex to handle when using flash memory.

Why we need a **special** filesystem?

1. RAM is limited, hence conventional buffers and inode data is too much.

2. Write operations for flash memory are different from magnetic-disk operations.

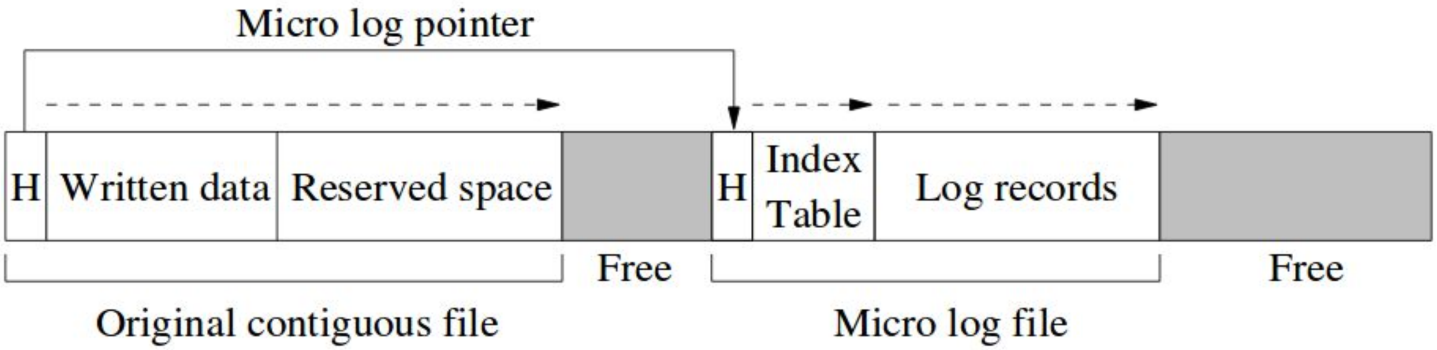
We do not want fragmentation.

We do not want small modifications.

Coffee File System

Append-only files are stored in the simplest way as contiguous group of pages.

Once a file is modified, Coffee creates an accompanying micro log structure and links it with the file.



- ★ Read and append speed is over 92% of the direct flash driver speed.
- ★ Constant RAM footprint for an open file.