University of BRISTOL

Computer Systems B

COMS20012

Introduction to Operating Systems and Security

bristol.ac.uk

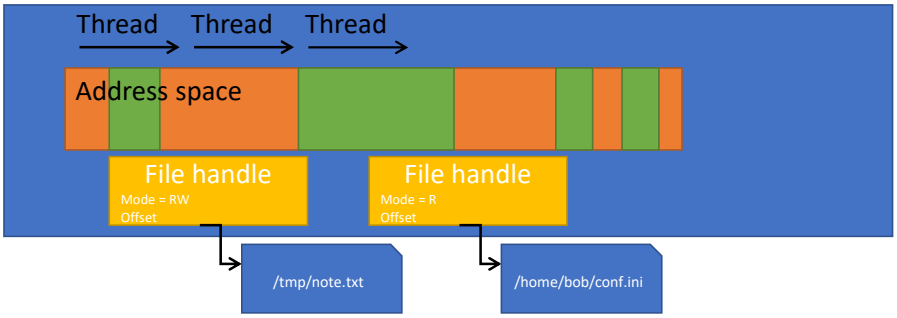
University of BRISTOL

File handles

bristol.ac.uk

Previously in COMS20012

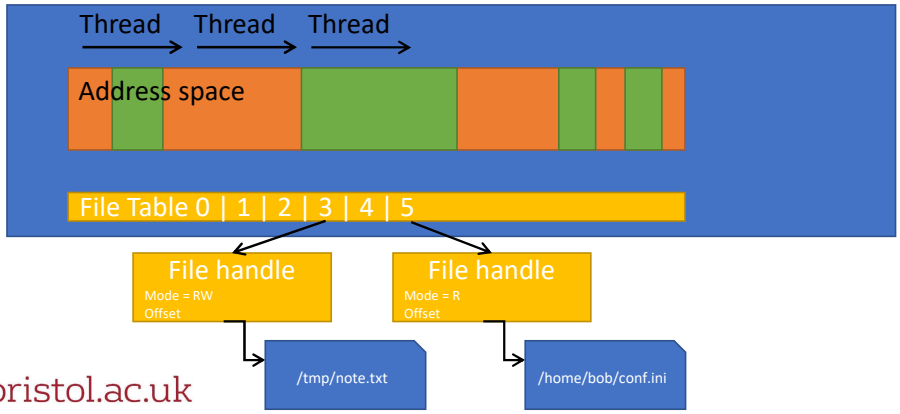
- Processes abstraction



bristol.ac.uk

Revised process model

- How process use files is important for today



bristol.ac.uk

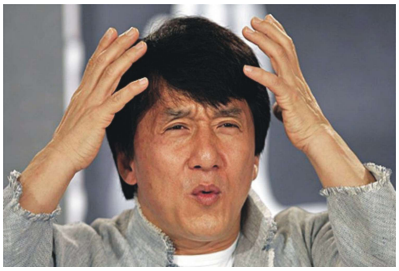
File handles

- **File descriptor** an integer representing the index of a file in the file table.
  - Value returned by the *open* system call.
- This integer correspond to a **file handle** maintained by the kernel
- Which reference a **file object** also maintained by the kernel
- The file object is then mapped to **blocks** on disk
- Three level of indirection
  - File descriptor -> file handle
  - File handle -> file object
  - File object -> blocks

bristol.ac.uk

File handles

- **File descriptor** an integer representing the index of a file in the file table.
  - Value returned by the *open* system call.
- This integer correspond to a **file handle** maintained by the kernel
- Which reference a **file object** also maintained by the kernel
- The file object is then mapped to **blocks** on disk
- Three level of indirection
  - File descriptor -> file handle
  - File handle -> file object
  - File object -> blocks
- **Why?**



bristol.ac.uk

Sharing file states

- The additional levels of indirection allows file states to be shared separately
- **File descriptors** are **private** to each process
- **File handles** are **private** at process creation but **can be shared**.
  - Store file **offset** were next read/write will take place
  - Can be **deliberately shared** between processes
- File objects hold other file states and is **shared transparently** between processes

bristol.ac.uk

File systems

- To be continued in **Week 9**

bristol.ac.uk

University of BRISTOL

Thank you

bristol.ac.uk