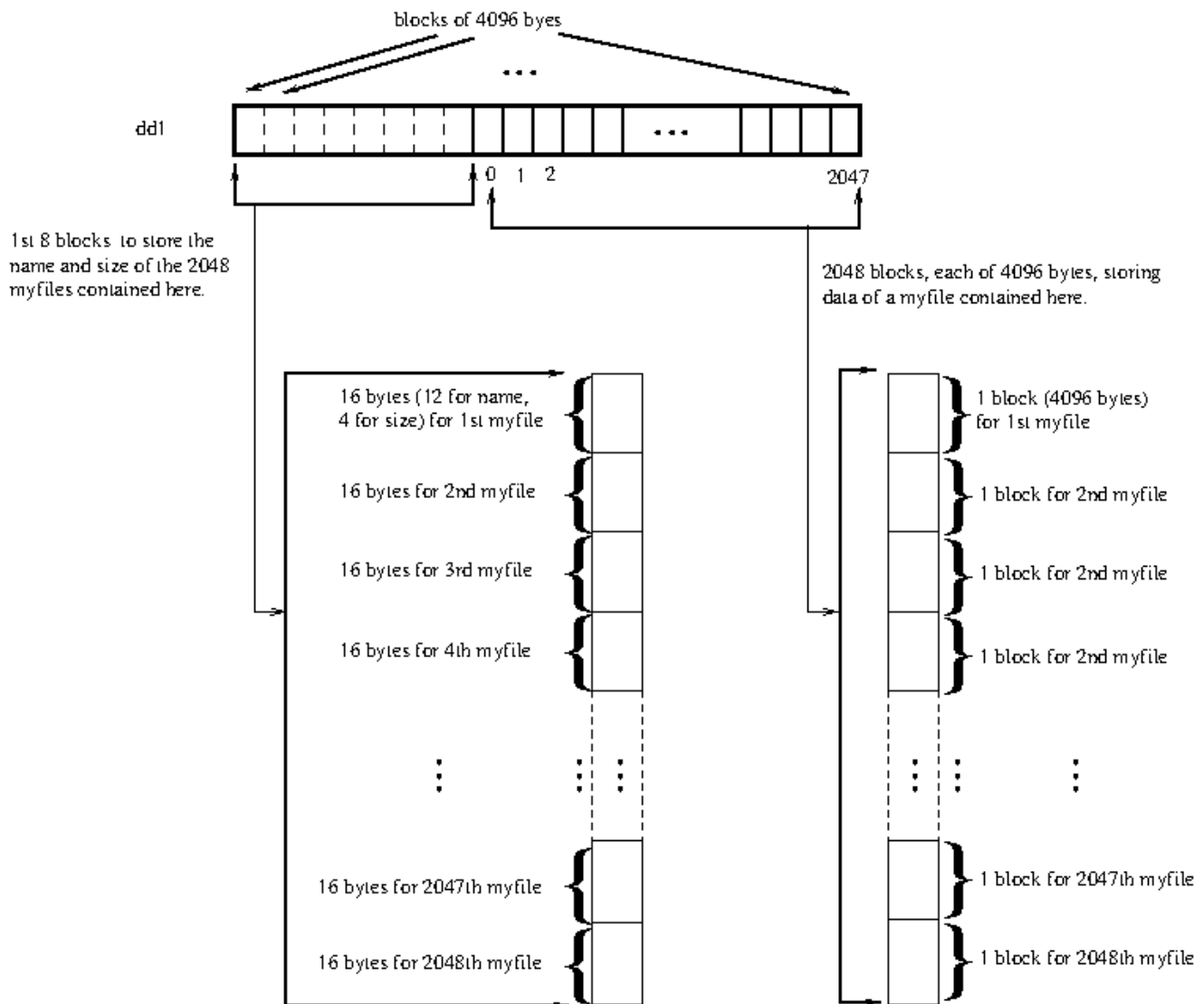


# Operating System Laboratory

## Assignment Statement for March 18, 2025

In the previous week a trivial filesystem (myfs) was implemented in a regular Linux file (dd1). The organization of myfs is depicted in the following figure.



The key points (and limitations) of myfs is given below.

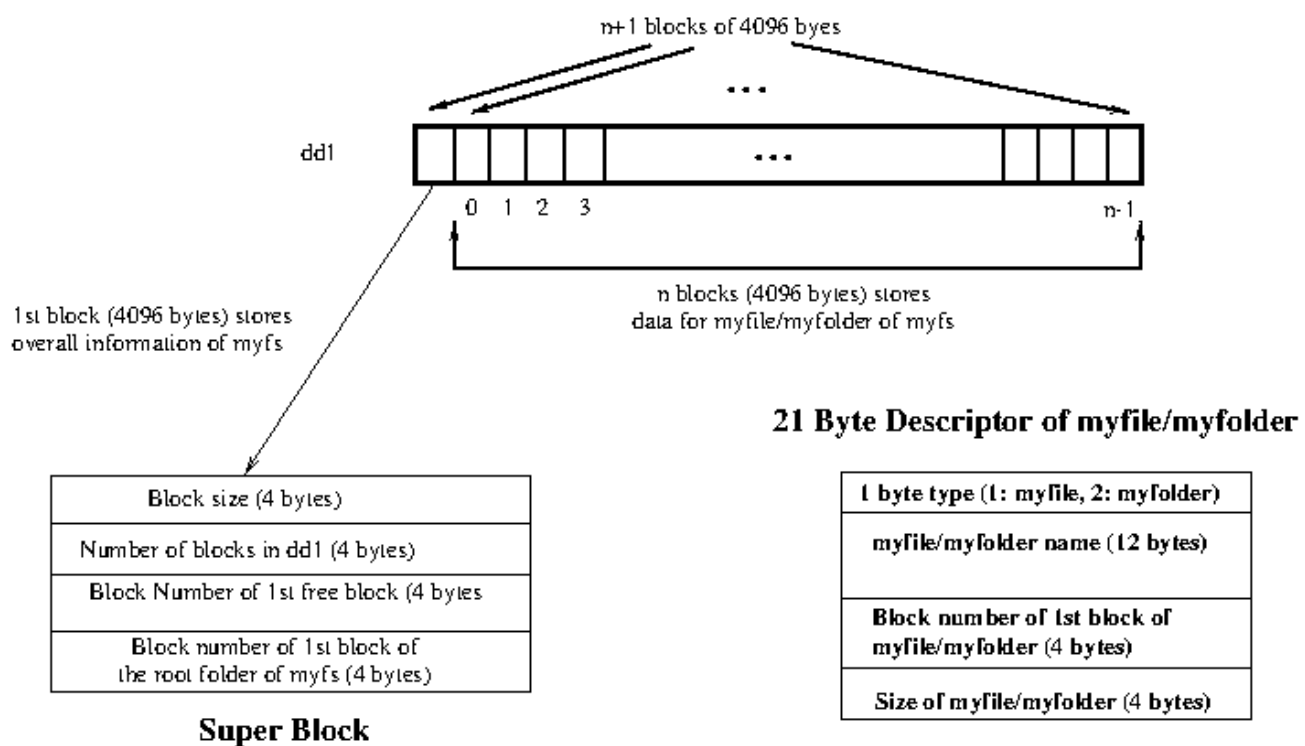
- I. Size of dd1 was fixed (2056 blocks, each of 4096 bytes)
- II. There was no idea of folder, it was a flat filesystem.
- III. myfs can have at most 2048 myfiles.
- IV. Each myfile in myfs could have at most 4096 bytes.
- V. myfile names in myfs can be of maximum 12 characters.

- VI. 4 functions were provided on myfs:
- mymkfs <Linux File> {Eg., mymkfs dd1}
  - mycopyto
  - mycopyfrom
  - myrm

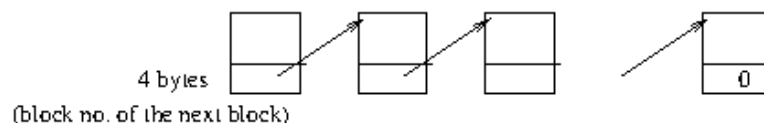
Today's assignment is an extension of the above mentioned assignment supporting the following features (overriding previous version of myfs).

- dd1 can have any number of blocks.
- Folders (myfolder) are supported.
- myfiles can be of any size
- myfolder can contain any number of sub myfolders, and/or myfiles.

The organization of myfs, now, will be as shown in the following figure.



**myfile/myfolder data blocks, and free blocks in myfs are kept as chains**



Please note that myfile (myfolder) needs multiple blocks. These blocks will be maintained as a chain, that is, the last 4 bytes of a block would contain the block number of the next block of myfile (myfolder). myfile descriptor now have one additional field (4 byte), block number of the 1<sup>st</sup> block of

the myfile (myfolder). The data part of a myfolder will be collection of 21 byte descriptors of the myfiles or submyfolders that the myfolder contains.

Like in the previous assignment, you have to implement 4 functions for this extended myfs

E. `mymkfs <Linux File> {Eg., mymkfs dd1}`

F. `mycopyto`

G. `mycopyfrom`

H. `myrm`