Solutions for the First Quiz

COSC 6360

Fall 2014

Questions with short answers

- What is the main advantage of ***mapped files***?

Questions with short answers

- What is the main advantage of ***mapped files***?
  - *Mapped files reduce the number of context switches in the file subsystem.*

Questions with short answers

- What is the main advantage of ***cylinder groups***?

Questions with short answers

- What is the main advantage of ***cylinder groups***?
  - *Since each cylinder group can contain both i-nodes and data blocks, the contents of most files can be located in the same cylinder group as their i-node.*

Questions with short answers

- Why does FFS use ***blocking writes*** for all its metadata updates?

Questions with short answers

- Why does FFS use ***blocking writes*** for all its metadata updates?
    - *To ensure that all metadata updates are written to disk in the right order.*

Questions with short answers

- In MACH, what should be the ***inheritance attribute*** of an address range that contains a ***mapped file***?

Questions with short answers

- In MACH, what should be the ***inheritance attribute*** of an address range that contains a ***mapped file***?

  - ***Shared***

Questions with short answers

- What does the UNIX ***set-userid bit*** allow?

Questions with short answers

- What does the UNIX ***set-userid bit*** allow?

  - *It allows a program to be executed with the* **rights of its owner** *rather than with the rights of the user executing it.*

Questions with short answers

- Where does UNIX store file ***access control lists***?

  Hint: *you probably know them under a different name.*

Questions with short answers

- Where does UNIX store file ***access control lists***?

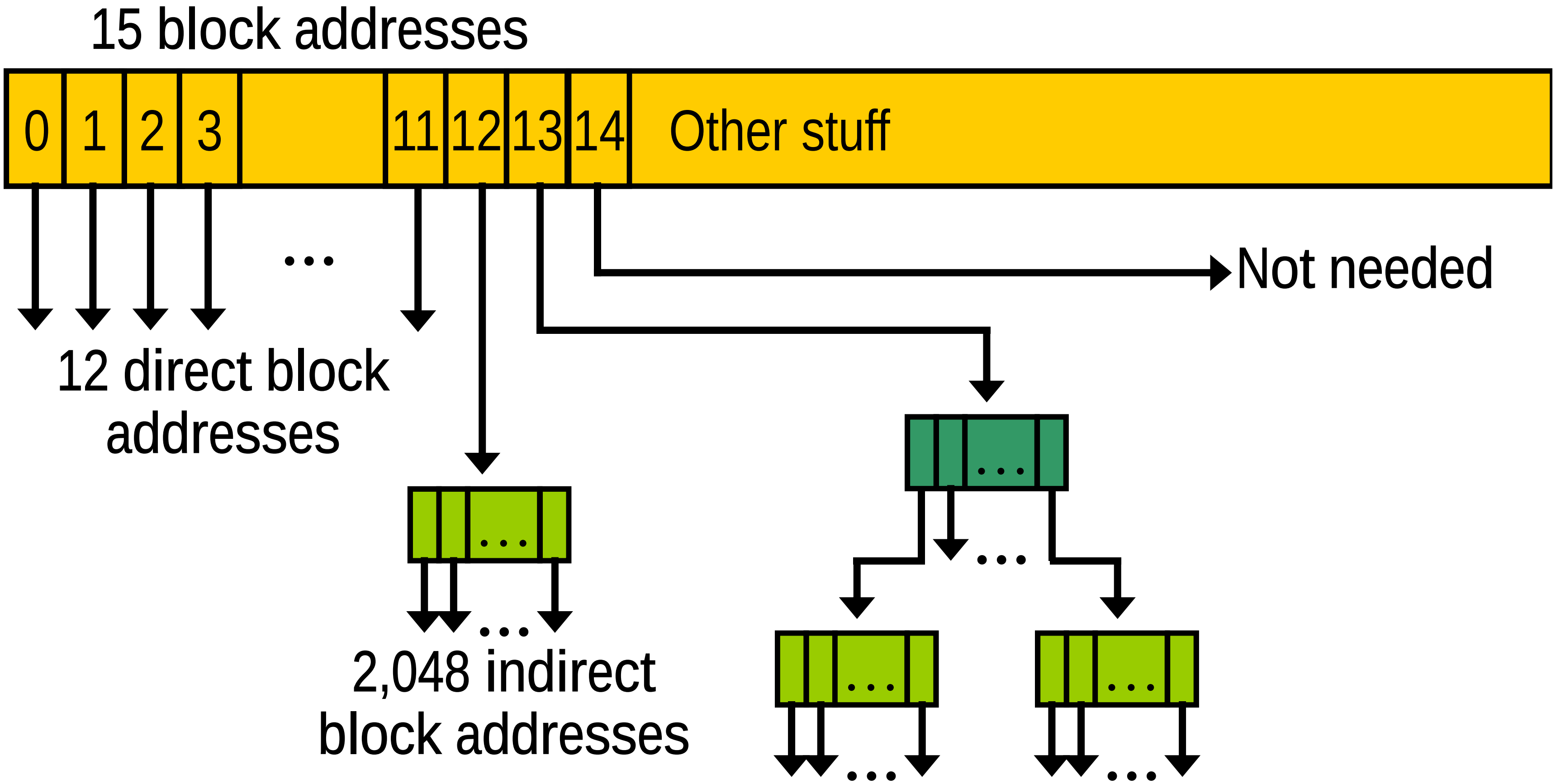
  Hint: *you probably know them under a different name.*

  - *In the* ***i-node*** *of each file*

File system

- A 32-bit UNIX file system has 8 kilobyte blocks and i-nodes with 15 block addresses. How many file ***blocks*** can be accessed (4×5 points)
  - □ Directly from the i-node?
  - □ With one level of indirection?
  - □ With two levels of indirections?
  - □ With three levels of indirections?

File System



15 block addresses

0 1 2 3 … 11 12 13 14 Other stuff

…

Not needed

12 direct block addresses

…

2,048 indirect block addresses

<u>Block size =8 KB</u>

<u>32-bit addresses</u>

1,024×1,024 double indirect block addresses
but we only use 512K – 2,048 – 12 of them

File system

- A 32-bit UNIX file system has 8 kilobyte blocks and i-nodes with 15 block addresses. How many file ***blocks*** can be accessed (4×5 points)
  - □ Directly from the i-node? **12 *blocks***
  - □ With one level of indirection?
  - □ With two levels of indirections?
  - □ With three levels of indirections?

File system

- A 32-bit UNIX file system has 8 kilobyte blocks and i-nodes with 15 block addresses. How many file ***blocks*** can be accessed (4×5 points)
    - □ Directly from the i-node? **12 *blocks***
    - □ With one level of indirection? **2,048 *blocks***
    - □ With two levels of indirections?
    - □ With three levels of indirections?

File system

- A 32-bit UNIX file system has 8 kilobyte blocks and i-nodes with 15 block addresses. How many file ***blocks*** can be accessed (4×5 points)
  - □ Directly from the i-node? **12 *blocks***
  - □ With one level of indirection? **2,048 *blocks***
  - □ With two levels of indirections?
    512K – 2K – 12 *blocks*
  - □ With three levels of indirections?

File system

- A 32-bit UNIX file system has 8 kilobyte blocks and i-nodes with 15 block addresses. How many file ***blocks*** can be accessed (4×5 points)
  - □ Directly from the i-node? **12 *blocks***
  - □ With one level of indirection? **2,048 *blocks***
  - □ With two levels of indirections?

    512K – 2K – 12 *blocks*
  - □ With three levels of indirections? **0 *blocks***

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
  - What is the function of the ***first hand*** of that clock? (5 easy points)

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
  - □ What is the function of the ***first hand*** of that clock? (5 easy points)
    - *It* **invalidates** *all the pages it touches.*

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
  - What is the function of the ***second hand*** of the clock? (5 easy points)

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
  - □ What is the function of the ***second hand*** of the clock? (5 easy points)
    - *It* **expels** *from memory all the invalid pages it touches.*

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
  - What would happen if the second hand follows ***too closely*** the first one? (5 points)

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
  - What would happen if the second hand follows ***too closely*** the first one? (5 points)
    - *Pages would be expelled too quickly*
    - *There will be too many page faults*

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
    - What would happen if the second hand is ***too far apart*** from the first one? (5 points)

Virtual memory

- Consider the ***Two-Handed Clock*** policy of Berkeley UNIX, where the page-referenced bit is simulated by software.
  - What would happen if the second hand is ***too far apart*** from the first one? (5 points)
    - *Pages would remain too long in main memory after their last access.*
    - *It would be more difficult to find free page frames.*