DATA STRUCTURES KERNEL USES TO HANDLE OPEN FILES IN LINUX

There are three data structures (or tables) kernel uses to handle open files.

Process Table Entry: Every process has a table called : Process table. When a process opens a file, a file descriptor is created and returned. And, an entry is created in this table. Each entry in this table contains flags related to the file descriptor and a pointer to a file table entry.

File Table : Kernel maintains a table of all open files system wide. This is not just process wide, but system wide. Each entry in this table contains : file status such as read, write, append mode, current file offset and a pointer to the v-node table for the file.

V-node Table: Each entry in this table contains information about each open file, information about the type of file, pointers to functions that operate on the file. The information also contains the i-node of the file. The i-node contains owner of the file, size of the file, pointer to where the actual data blocks for the file are located on disk.

|  |
| --- |
| Two different processes open three different files |

|  |
| --- |
| Two different processes open the same file.  http://poincare.matf.bg.ac.rs/~ivana/courses/ps/sistemi_knjige/pomocno/apue/APUE/0201433079/images/0201433079/graphics/03fig07.gif;423615 |

|  |
| --- |
| A parent process opens a file and forks a child. Then , parent and child have the same file table entries |

With these data structures, we will discuss what happens when certain operators are performed on a file.

When a write is done on a file, the current file offset in the file table entry is incremented by the number of bytes written. If this causes the current file offset to exceed the current file size, the current file size is updated in the v-node structure.

When a file is opened with the O\_APPEND flag, the corresponding flag is set in the file table. For every write to the file, current file size is updated in the v-node structure