

C. Combining The Teachings Of Kenley And Long

~~A POSITA would have been motivated, with~~⁴⁹. As discussed below and herein, there was a motivation to combine Kenley and Long, as well as a reasonable expectation of success to implement Kenley's file-based backup system within doing so. Specifically, a POSITA would have been motivated to use the modification-driven archival approach taught by Long, to for a file-based backup system as taught by Kenley because backing up files when they are modified would improve the reliability and provide other benefits to of the file-based backup system.

50. For example, a POSITA looking to implement the "incremental backup" technique of Kenley provides would have recognized that the incremental backup is only "[a] copy of ... specified files of data that has been modified since a specified date, usually the date of the last full or incremental backup." E.g., EX1014, 7:19-30, 5:3-6. EX1014, 7:19-30 (discussing use of the incremental backup), 5:3-6 (defining incremental backup). Kenley explains that the incremental backup process "greatly reduc[es] the amount of data that must be copied" which "vastly reduce[s] the amount of time required to back up" files. EX1014, 12:66-13:17. Long, in turn, notes the data risks posed by power outages and operator error (EX1004, 1:46-55), and provides "significant improvements in hard disk error recovery" (*id.*, 1:62-65). These improvements include increased reliability by creating backup data after each modification, thus avoiding the potential loss of data created in between the timed

backups of Kenley and other ~~similar systems. *Id.*, 4:23–27. This combination~~
~~both incremental backup systems. *Id.*, 4:23-27. The POSITA would have been~~
~~motivated to incorporate Long’s technique of backing up data upon modification~~
~~into a file-based backup system such as Kenley because the Long technique further~~
~~increases the reliability of file backup, as there is no delay between modification and~~
~~backup. This combination~~ serves the goal of “reduc[ing] the time associated with
conventional data migration schemes” taught by Kenley, ~~and provides which is~~
~~consistent with the goal of providing~~ an “efficient, reliable and cost[-]effective
architecture for disk fault tolerance” ~~as~~ taught by Long. EX1014, 2:26-30; EX1004,
2:13-14. ~~As explained in the following paragraphs, for at least the following reasons,~~
~~a POSITA would have made this combination. EX1003, ¶¶49–51.~~

51. For these and the following additional reasons, a POSITA would have
been motivated with a reasonable expectation of success to implement a file-based
system such as that described by Kenley with Long’s backup and archival
techniques.

52. First, Long teaches a “Fault Tolerant Computer With Archival Rollback
Capabilities” that “includes significant improvements in hard disk error recovery”
and “provides a[n] efficient, reliable and cost[-]effective architecture for disk fault
tolerance.” EX1004, 1:2-3, 1:62-64, 2:13-14. A POSITA would have thus been
motivated to improve Kenley’s file-based “hierarchical digital data storage system”

(EX1014, 2:33-36, 3:1-47) by adding Long's event-driven backup process that efficiently saved backups prior to any modifications of a file stored on hard disk. ~~EX1003, ¶52.~~

53. Second, a POSITA ~~knew~~would have known that ~~Long's teachings were applicable to files. Long expressly states that its "invention is applicable to any mass storage media wherein old data may be overwritten with new data" and files. While Long broadly discusses "data on a hard disk" and the copying of such data prior to modifying "the contents of the hard disk."~~Long does not specifically consider data stored in files. EX1004, ~~10:54-56,~~ 1:46-48, 4:23-25, 7:28-29. ~~Thus~~However, a POSITA reading Long, with its discussion of applications modifying data, would immediately contemplate use of ~~its technique~~the Long approach within file-based systems, for preserving data stored in files within that system. ~~See also EX1011, 19 ("Backup software can operate either on files or on physical disk blocks."); EX1005, 1:32-34 (the "prior state of the art" included archive systems that "maintain[ed] a complete history of the changes made to ... a particular file").~~ The POSITA knew that data on a hard disk was most commonly organized as files to facilitate data access using a file system. *E.g.*, EX1013, 5:18-51; ~~EX1012, 1:27-60.~~ (describing the File Allocation Table (FAT) file system created for the MS-DOS operating system); EX1012, 1:27-60 (explaining different file systems for different operating systems such as MS-DOS, Macintosh, and UNIX); supra Section VII. Indeed, the

background section of one background reference flatly states what was well-known: “[i]nformation processed by a computer is organized into distinct units called files,” each of which contains related data.” EX1012, 1:26-27. Moreover, it was well-known that applications in file-based systems would perform write and other operations that modified the file. E.g., EX1010, 1:40-43. ~~This is reinforced by As Kenley, which describes~~ explains, “stor[ing] data organized as files” and using file attributes to track the backup location of files enables the data to be more easily and transparently recovered from archive storage media for use by an application. EX1014, 3:29-32, 7:19-62. ~~EX1003, ¶53. (explaining how the file-based approach enables the system to “properly backup and restore files”).~~

54. Third, the two references are complementary, as each teaches ~~backup systems with~~ at least three tiers of storage—primary storage (e.g., RAM or semiconductor memory buffer), secondary storage (e.g., hard disk), and archive or backing storage (e.g., removable or external storage devices)—for use in working with and backing up data. EX1014, 2:44-55; (describing three tiers of storage); EX1004, 7:43-53 (describing first copying data to a semiconductor memory buffer and then writing the data to audit buffer on disk), 8:29-9:9 (describing moving the data from the audit buffer on disk to archive media). Kenley also teaches “‘transparent’ access to digital data” stored in archive storage media, with “access that entails no significant delay in ongoing processes” and “a storage system

interface that communicates with applications programs in a manner functionally identical to that of the existing system interface.” EX1014, 2:33-43. This “transparent” archiving aligns with Long’s teachings on a system that “allows the archiving to take place in the background so that the user may continue processing during an archive operation.” EX1004, 8:42-49; ~~EX1003, ¶54.~~

55. Fourth, Kenley’s teaching of a database-held system catalog that “stor[es] and provid[es] transparent access to catalogued file attribute data” (EX1014, 3:17-24) dovetails with Long’s teachings on an audit header for similarly maintaining data on the current location of backup copies of data (e.g., EX1004, 7:47-53). Kenley’s system catalog “enhances transparent access to file attribute data, independent of whether the file is stored in the secondary storage element or has been previously migrated from the secondary storage element” to the backing store. EX1014, 3:25-28. Thus, a POSITA would naturally augment the catalog database of Kenley to store relevant information from the audit header taught by Long, thus “enhanc[ing] transparent access.” *Id.*; see ~~also~~ EX1004, 7:47-53, 9:7-9 (describing updating the audit header to reflect changes after migrating data); EX1014, 6:35-40 (describing the system catalog ~~is as~~ a part of a relational database), 8:26-29 (describing updating the database with a catalog of all files backed up ~~files~~), 10:43-52 (describing using the ~~database~~ backup catalogs in the database to locate backup files); ~~EX1003, ¶55.).~~

56. Fifth, Long teaches a straightforward modification-driven archival approach that applied to known multi-tier backup—archival systems. EX1004, 11:54- 56: (“The present invention is applicable to any mass storage media wherein old data may be overwritten with new data.”). A POSITA would have easily applied that approach in a file-based system such as Kenley’s. As explained in mapping the claims below, this method uses the same event-driven approach as the Challenged Claims. *Infra* ~~Section~~Sections VIII.D.C.1-VIII.C.23. Thus, a POSITA implementing a file- based backup system such as that taught by Kenley, adapted with Long’s archival method, had a reasonable expectation of success in implementing the methods of the Challenged Claims. ~~EX1003, ¶56.~~

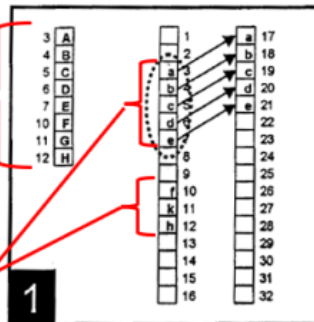
57. The reasonable expectation of success, and overall obviousness, of implementing Long’s technique for a file-based system such as Kenley’s is confirmed by another ~~1998 filed~~ prior-art patent, U.S. 6,016,553 (EX1016; “Schneider-US”) that expressly describes ~~using~~use of Long’s audit buffer technique ~~to save files for saving a file’s data before alteration; the file is altered.~~ EX1016, 46:36-47:8. ~~That background reference reflects the knowledge of a POSITA and shows in FIG. 37 (using file overwrite operation to explain “what normally happens when data is written.”~~ *Id.* ~~It” and then contrasts that with~~describing how the Long technique (~~aka~~referred to as the “Move” method) ~~in FIG. 38, which saves a file just before an overwrite. Id.; prior to it being overwritten); see also id., 46:25-26;~~

(discussing “fundamental methods for saving the prior states of overwritten data”);
id., 6:33-36 (presenting “fundamental methods of recording the original state of
information prior to its being altered”). In reference to FIG. 38 and related figures
that illustrate the “Move” method, Schneider-US explains that “[t]he column on the
left represents a file” (id., 46:39-40), the column in the middle “represents the OS-
visible area” of the hard disk, and the column on the right “is a history buffer visible
only to the engine” (id., 46:63-47:1). Schneider- US then explains how, before the
file overwrite is carried out on the hard disk (middle column), the system copies the
file’s contents into the history buffer. Id., 47:1-9. Schneider-US assumes a POSITA
would understand Long’s method can be readily applied to file data, because
Schneider-US does not indicate any changes to Long in order to be able to do so.

FRAME 1

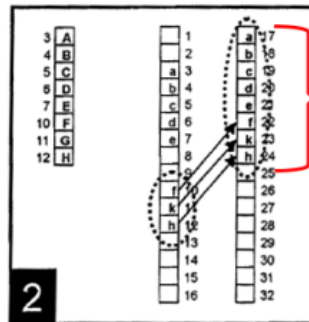
file, as updated
in RAM

file on disk's active
portion, prior to
overwriting with
updated file

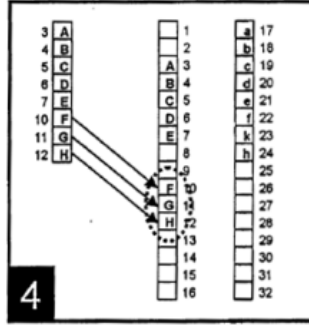
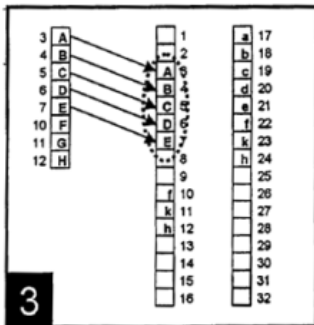


FRAME 2

pre-update file,
moved from
active disk to
audit buffer (aka
"history buffer")



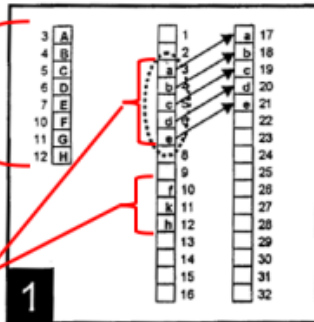
FRAMES 3-4:
updated file
overwrites pre-
update file on
disk's active
portion



FRAME 1

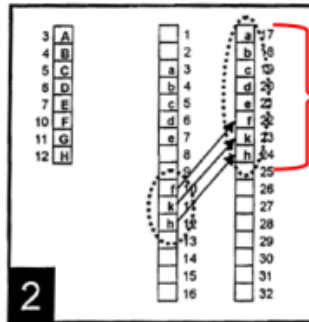
file, as updated
in RAM

file on disk's active
portion, prior to
overwriting with
updated file

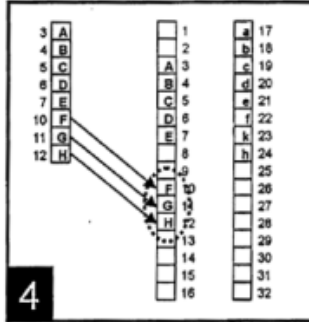
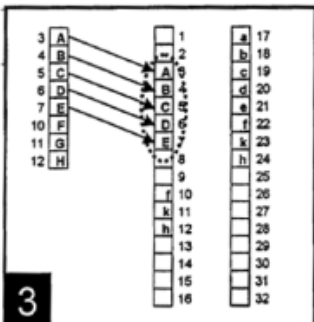


FRAME 2

pre-update file,
moved from
active disk to
audit buffer (aka
"history buffer")



FRAMES 3-4:
updated file
overwrites pre-
update file on
disk's active
portion



Id., FIG. 38 (annotated based on description at 46:36-47:9). ~~This confirms a POSITA~~
~~58. Thus, it~~ would have ~~naturally and easily implemented Long's method for been~~
~~obvious to implement a~~ file-based systems. EX1003, ¶57.

~~system as taught by Kenley using Long's data backup and archiving method.~~

So implemented, the resulting Kenley-Long system would, based on computer-executable code within the system, ~~perform at least these operations:~~

- Generate a modification request, as taught by Long, any time the operating system or a user application “need[ed] to modify the contents of” a file stored on the hard disk. For example, a text-editing program issuing a write operation for a file would be received by the computer system and then “sent to the audit subsystem.” EX1004, 7:27-53.
- Carry out the audit and archive technique taught by Long, on the file being modified, including the copying of files first to the memory buffer, second to the audit buffer portion of the hard drive, and then to external storage, as taught by Long, thus creating “archive copies” of files ~~within a filed based system~~ as taught by Kenley. EX1014, Abstract.
- Maintain relevant audit header information, as taught by Long, in a backup catalog in a relational database, as taught by Kenley. EX1003, ¶¶~~Supra ¶55,~~
~~58.~~