EXHIBIT 1B

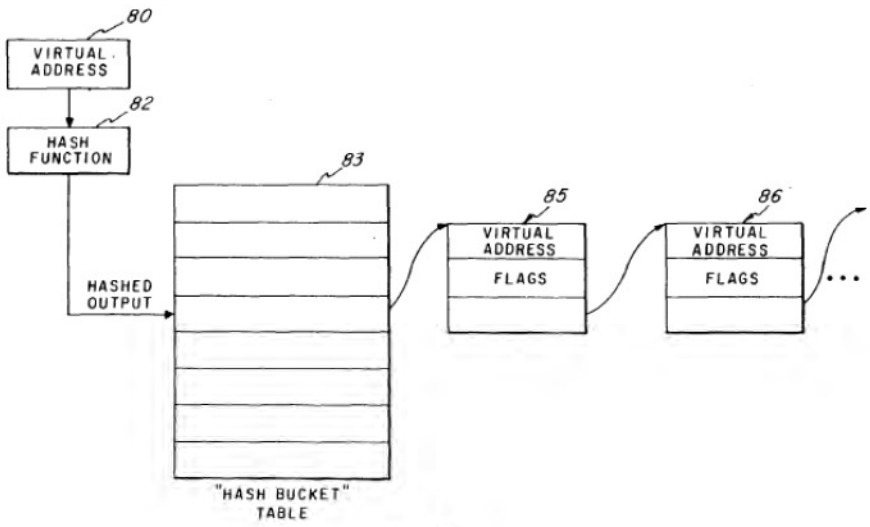

Fig. 7

Thatte at Figure 7

the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed, and

Thatte discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is searched. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular, Thatte discloses that the pointers to the records in the hash table of linked lists are either marked or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18. According to Thatte, this identifying and removing of expired records occurs while the linked list is accessed in the insertion operation. Thatte at Col. 7, lines 21-26.

means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.

Thatte discloses means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore evacuate the unmarked garbage records in the hash table

of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

CLAIM 2

The information storage and retrieval system according to claim 1 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Thatte discloses an information storage and retrieval system according to claim 1 further including means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 2, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

CLAIM 3

A method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring, the method comprising the steps of:

Thatte discloses a method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring.

Thatte at Col. 8, lines 49-62; Col. 5, lines 40-62; Col. 7, lines 1-60. In particular, discloses using a linked list to store, retrieve, and otherwise access these records utilizing insert, delete, reconcile, and locate operations. Thatte at Col. 8, lines 57-59; Col. 6, lines 64-67; Col. 8, lines 49-62. Thatte further discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records as garbage. Thatte at Col. 7, line 61- Col. 8, line 18.

accessing the linked list of records,

Thatte discloses accessing the linked list of records, utilizing insert, delete, reconcile, and locate operations. Thatte at Col. 8, lines 57-59; Col. 6, lines 64-67; Col. 8, lines 49-62.

identifying at least some of the automatically expired ones of the records, and

Thatte discloses identifying at least some of the automatically expired ones of the records. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular, Thatte discloses that the pointers to the records in the hash table of linked lists are either marked or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18.

removing at least some of the automatically expired records from the linked list when the linked list is accessed.

Thatte discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore expire and evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

CLAIM 4

The method according to claim 3 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Thatte discloses the method according to claim 3 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In

particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 4, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

CLAIM 5

An information storage and retrieval system, the system comprising:

Thatte discloses an information storage and retrieval system. Thatte at Col. 1, lines 7-13; Col. 2, lines 3- 10; Col. 5, lines 40-62. In particular, Thatte discloses a memory management unit ("MMU") and associated reference count filter to perform memory management functions. Thatte at Col. 2, lines 3-10; Col. 6, lines 1-10; Col. 6, lines 64-67; Col. 5, line 66- Col. 6, line 1.

a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring,

Thatte discloses a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with the same hash address, at least some of the records automatically expiring. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Col. 7, line 61- Col. 8, line 18; Fig. 7. In particular, Thatte discloses that the reference count filter that stores the records is implemented by means of a hash table, which "can efficiently support the insert, delete, and reconcile operations" used to access the reference count

filter. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Fig. 7. According to the '120 patent, in external chaining, "each hash table location is a pointer to the head of a linked list of records, all of whose keys translate under the hashing function to that very hash table address." '120 patent at Col. 1, lines 58-62. Thatte further discloses using an external chaining technique in that "[t]he hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located;" this located entry "may be a pointer which may point to a linked list, etc. of virtual addresses" where the records have the same hash address. Thatte at Col. 8, lines 39-62. Additionally, Thatte discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records in the linked list as garbage. Thatte at Col. 7, line 61-Col. 8, line 18.

a record search means utilizing a search key to access a linked list of records having the same hash address,

Thatte discloses a record search means, the "locate" operation of the MMU and reference count filter, utilizing a search key to access a linked list of records having the same hash address. Thatte at Col. 8, lines 39-62; Fig. 7. Specifically, Thatte discloses that the MMU in association with the reference count filter performs insert, delete, reconcile, and locate operations. Thatte at Col. 6, lines 64-68; Col. 8, line 55-62. As further disclosed in Thatte, a virtual address functions as a search key, as it is "applied to means for implementing a hash function," wherein the "hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located" in a linked list, therefore having the same hash address. Thatte at Col. 8, lines 49-62; Fig. 7.

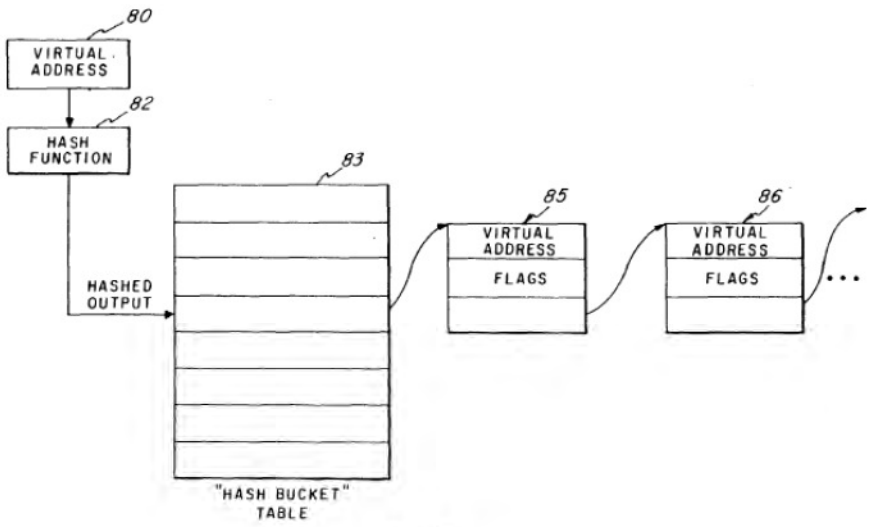

Fig. 7

Thatte at Figure 7

the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed, and

Thatte discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular, Thatte discloses that the pointers to the records in the hash table of linked lists are either marked or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18. According to Thatte, this identifying and removing of expired records occurs while the linked list is accessed in the insertion operation. Thatte at Col. 7, lines 21-26.

mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.

Thatte discloses means, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some of the expired ones of the records in the linked list. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Under the broadest reasonable interpretation, the use

of "and" in the phrase "inserting, retrieving, and deleting records" is ambiguous, and thus this limitation can reasonably be interpreted to mean "inserting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records." While performing the insertion operation for example, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

CLAIM 6

The information storage and retrieval system according to claim 5 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Thatte discloses the information storage and retrieval system according to claim 5 further including means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 6, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

CLAIM 7

A method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring, the method comprising the steps of:

Thatte discloses a method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with the same hash address, at least some of the records automatically expiring. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Col. 7, line 61- Col. 8, line 18; Fig. 7. In particular, Thatte discloses that the reference count filter that stores the records is implemented by means of a hash table, which "can efficiently support the insert, delete, and reconcile operations" used to access the reference count filter. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Fig. 7. According to the '120 patent, in external chaining, "each hash table location is a pointer to the head of a linked list of records, all of whose keys translate under the hashing function to that very hash table address." '120 patent at Col. 1, lines 58-62. Thatte further discloses using an external chaining technique in that "[t]he hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located;" this located entry "may be a pointer which may point to a linked list, etc. of virtual addresses" where the records have the same hash address. Thatte at Col. 8, lines 39-62; Fig. 7.

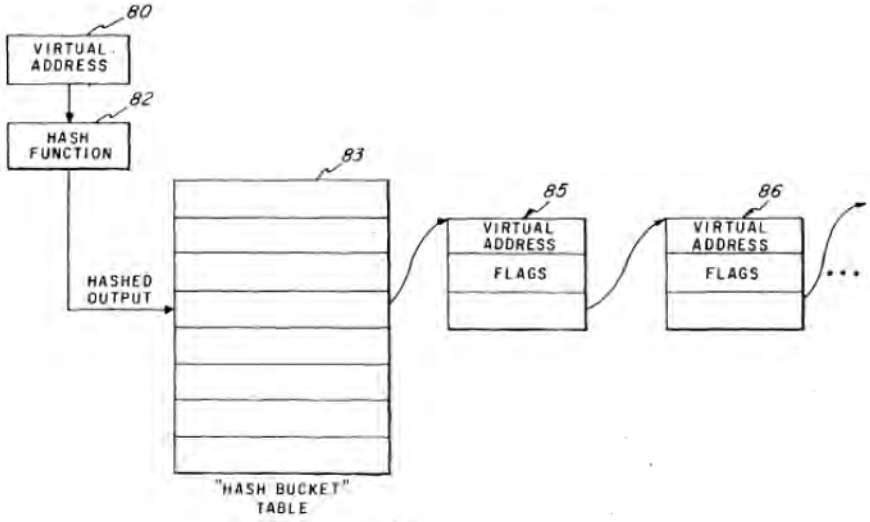

Fig. 7

Thatte at Figure 7

Additionally, Thatte discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records in the linked list as garbage. Thatte at Col. 7, line 61- Col. 8, line 18.

accessing a linked list of records having same hash address,

Thatte discloses accessing a linked list of records having the same hash address. Thatte at Col. 8, lines 39-62; Fig. 7. Specifically, Thatte discloses that the MMU in association with the reference count filter performs insert, delete, reconcile, and locate operations to access the linked list. Thatte at Col. 6, lines 64-68; Col. 8, line 55-62. As further disclosed in Thatte, a virtual address functions as a search key, as it is "applied to means for implementing a hash function," wherein the "hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located" in a linked list, therefore having the same hash address. Thatte at Col. 8, lines 49-62; Fig. 7.

identifying at least some of the automatically expired ones of the records,

Thatte discloses identifying at least some of the automatically expired ones of the records. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular, Thatte discloses that the pointers to the records in the hash table of linked lists are either marked or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18.

removing at least some of the automatically expired records from the linked list when the linked list is accessed, and

Thatte discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore expire and evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

inserting, retrieving or deleting one of the records from the system following the step of removing.

Thatte discloses inserting, retrieving or deleting one of the records from the system following the step of removing. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore expire and evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18. Thatte further discloses that after removing the expired records in the reconciliation operation, the insertion operation can be completed. Thatte at Col. 7, lines 19-26.

CLAIM 8

The method according to claim 7 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Thatte discloses the method according to claim 7 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 8, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

D. CLAIMS 1-8 ARE RENDERED OBVIOUS BY THATTE IN VIEW OF DIRKS UNDER 35 U.S.C. § 103

Please see the attached Exhibit CC-D presenting claim charts for comparison of Thatte in view of Dirks with claims 1-8 of the '120 patent.

Reasons to Combine:

One of ordinary skill in the art would have been motivated to combine the memory manager in Dirks with the memory management unit in Thatte in order to provide for efficiently determining the extent to which the automatically expiring records in Thatte should be removed. Both Thatte and Dirks are directed towards navigating a list-type data structure and removing inactive or expired records as efficiently as possible. Furthermore, both Thatte and Dirks disclose the use of a hashing technique in an effort to effectively allocate space within a memory system. Additionally, one of ordinary skill in the art would recognize that the result of combining Thatte with Dirks would be nothing more than the predictable use of prior art elements according to their established functions. The result would simply be the application of Dirks' recycle operation to the hash table of linked lists data structure disclosed in Thatte.

CLAIM 1

An information storage and retrieval system, the system comprising:

Thatte in view of Dirks discloses an information storage and retrieval system. Thatte at Col. 1, lines 7-13; Col. 2, lines 3- 10; Col. 5, lines 40-62. In particular, Thatte discloses a memory management unit ("MMU") and associated reference count filter to perform memory management functions. Thatte at Col. 2, lines 3-10; Col. 6, lines 1-10; Col. 6, lines 64-67; Col. 5, line 66- Col. 6, line 1.

a linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring,

Thatte in view of Dirks discloses a linked list to store and provide access to records that contain virtual addresses, at least some of the records automatically expiring. Thatte at Col. 8, lines 49-62; Col. 5, lines 40-62; Col. 7, lines 1-60. In particular, Thatte discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records as garbage.

Thatte at Col. 7, line 61- Col. 8, line 18. Thatte further discloses using a linked list to store these records. Thatte at Col. 8, lines 57-59.

a record search means utilizing a search key to access the linked list,

Thatte in view of Dirks discloses a record search means, the "locate" operation of the MMU and reference count filter, utilizing a search key to access the linked list. Thatte at Col. 8, lines 39-62; Fig. 7. Specifically, Thatte discloses that the MMU in association with the reference count filter performs insert, delete, reconcile, and locate operations. Thatte at Col. 6, lines 64-68; Col. 8, line 55-62. As further disclosed in Thatte, a virtual address functions as a search key, as it is "applied to means for implementing a hash function," wherein the "hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located" in a linked list. Thatte at Col. 8, lines 49-62; Fig. 7.

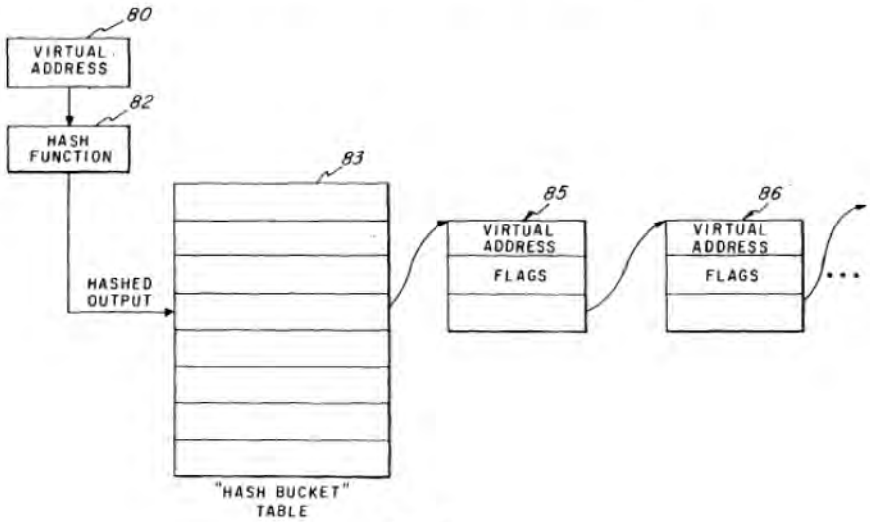

Fig. 7

Thatte at Figure 7

the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed, and

Thatte in view of Dirks discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is searched. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular, Thatte discloses that the pointers to the records in the hash table of linked lists are either marked

or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18. According to Thatte, this identifying and removing of expired records occurs while the linked list is accessed in the insertion operation. Thatte at Col. 7, lines 21-26.

means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.

Thatte in view of Dirks discloses means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

CLAIM 2

The information storage and retrieval system according to claim 1 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Thatte in view of Dirks discloses the information storage and retrieval system according to claim 1 further including the means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Thatte at Col. 2, lines 3-10; Col. 6, lines 1-10; Col. 6, lines 64-67; Col. 5, line 66- Col. 6, line 1; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Thatte discloses accessing a table of linked lists. Thatte at Col. 8, lines 59-62. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries k in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation k by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using

"[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 2, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 2, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 2.

CLAIM 3

A method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring, the method comprising the steps of:

Thatte in view of Dirks discloses a method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring. Thatte at Col. 8, lines 49-62; Col. 5, lines 40-62; Col. 7, lines 1-60. In particular, discloses using a linked list to store, retrieve, and otherwise access these records utilizing insert, delete, reconcile, and locate operations. Thatte at Col. 8, lines 57-59; Col. 6, lines 64-67; Col. 8, lines 49-62. Thatte further discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records as garbage. Thatte at Col. 7, line 61- Col. 8, line 18.

accessing the linked list of records,

Thatte in view of Dirks discloses accessing the linked list of records, utilizing insert, delete, reconcile, and locate operations. Thatte at Col. 8, lines 57-59; Col. 6, lines 64-67; Col. 8, lines 49-62.

identifying at least some of the automatically expired ones of the records, and

Thatte in view of Dirks discloses identifying at least some of the automatically expired ones of the records. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular,

Thatte discloses that the pointers to the records in the hash table of linked lists are either marked or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18.

removing at least some of the automatically expired records from the linked list when the linked list is accessed.

Thatte in view of Dirks discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore expire and evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

CLAIM 4

The method according to claim 3 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Thatte in view of Dirks discloses the method according to claim 3 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Thatte at Col. 8, lines 49-62; Col. 5, lines 40-62; Col. 7, lines 1-60; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Thatte discloses accessing a table of linked lists. Thatte at Col. 8, lines 59-62. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 4, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum

number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 4, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 4.

CLAIM 5

An information storage and retrieval system, the system comprising:

Thatte in view of Dirks discloses an information storage and retrieval system. Thatte at Col. 1, lines 7-13; Col. 2, lines 3- 10; Col. 5, lines 40-62. In particular, Thatte discloses a memory management unit ("MMU") and associated reference count filter to perform memory management functions. Thatte at Col. 2, lines 3-10; Col. 6, lines 1-10; Col. 6, lines 64-67; Col. 5, line 66- Col. 6, line 1.

a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring,

Thatte in view of Dirks discloses a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with the same hash address, at least some of the records automatically expiring. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Col. 7, line 61- Col. 8, line 18; Fig. 7. In particular, Thatte discloses that the reference count filter that stores the records is implemented by means of a hash table, which "can efficiently support the insert, delete, and reconcile operations" used to access the reference count filter. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Fig. 7. According to the '120 patent, in external chaining, "each hash table location is a pointer to the head of a linked list of records, all of whose keys translate under the hashing function to that very hash table address." '120 patent at Col. 1, lines 58-62. Thatte further discloses using an external chaining technique in that "[t]he hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located;" this located entry "may be a pointer which may point to a linked list, etc. of virtual addresses" where the records have the same hash

address. Thatte at Col. 8, lines 39-62. Additionally, Thatte discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records in the linked list as garbage. Thatte at Col. 7, line 61- Col. 8, line 18.

a record search means utilizing a search key to access a linked list of records having the same hash address,

Thatte in view of Dirks discloses a record search means, the "locate" operation of the MMU and reference count filter, utilizing a search key to access a linked list of records having the same hash address. Thatte at Col. 8, lines 39-62; Fig. 7. Specifically, Thatte discloses that the MMU in association with the reference count filter performs insert, delete, reconcile, and locate operations. Thatte at Col. 6, lines 64-68; Col. 8, line 55-62. As further disclosed in Thatte, a virtual address functions as a search key, as it is "applied to means for implementing a hash function," wherein the "hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located" in a linked list, therefore having the same hash address. Thatte at Col. 8, lines 49-62; Fig. 7.

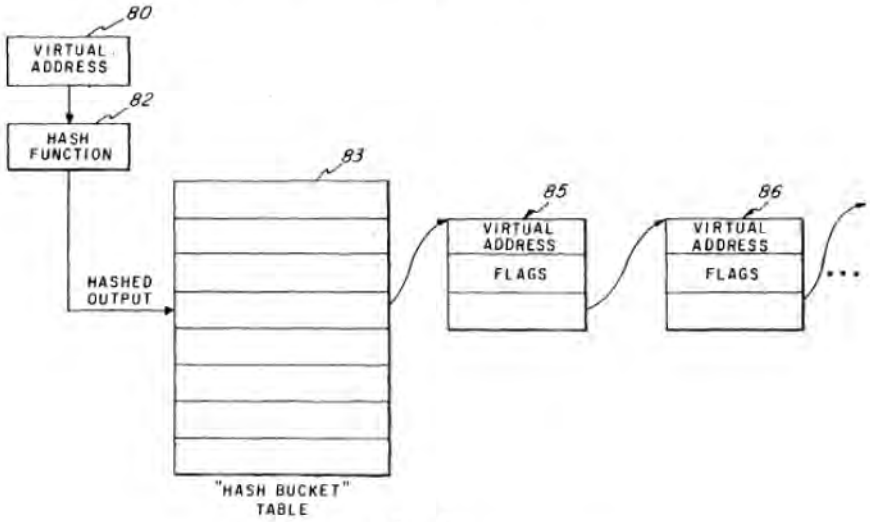

Fig. 7

Thatte at Figure 7

the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed, and

Thatte in view of Dirks discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular, Thatte discloses that the pointers to the records in the hash table of linked lists are either marked or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18. According to Thatte, this identifying and removing of expired records occurs while the linked list is accessed in the insertion operation. Thatte at Col. 7, lines 21-26.

mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.

Thatte in view of Dirks discloses means, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some of the expired ones of the records in the linked list. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Under the broadest reasonable interpretation, the use of "and" in the phrase "inserting, retrieving, and deleting records" is ambiguous, and thus this limitation can reasonably be interpreted to mean "inserting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records." While performing the insertion operation for example, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

CLAIM 6

The information storage and retrieval system according to claim 5 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Thatte in view of Dirks discloses the information storage and retrieval system according to claim 5 further including the means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Thatte at Col. 2, lines 3-10; Col. 6, lines 1-10; Col. 6, lines 64-67; Col. 5, line 66- Col. 6, line 1; Dirks at Col. 6, line 66- Col.

7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Thatte discloses accessing a table of linked lists. Thatte at Col. 8, lines 59-62. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 6, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 6, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 6.

CLAIM 7

A method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring, the method comprising the steps of:

Thatte in view of Dirks discloses a method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with the same hash address, at least some of the records automatically expiring. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Col. 7, line 61- Col. 8, line 18; Fig. 7. In particular, Thatte discloses that the reference count filter that stores the records is implemented by means of a hash table, which "can efficiently support the insert,

delete, and reconcile operations" used to access the reference count filter. Thatte at Col. 6, lines 48-63; Col. 8, lines 39-62; Fig. 7. According to the '120 patent, in external chaining, "each hash table location is a pointer to the head of a linked list of records, all of whose keys translate under the hashing function to that very hash table address." '120 patent at Col. 1, lines 58-62. Thatte further discloses using an external chaining technique in that "[t]he hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located;" this located entry "may be a pointer which may point to a linked list, etc. of virtual addresses" where the records have the same hash address. Thatte at Col. 8, lines 39-62; Fig. 7.

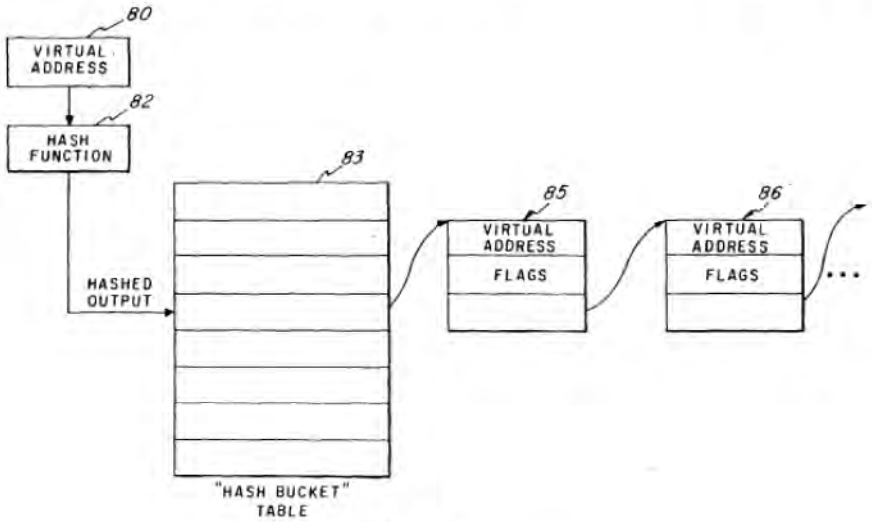

Fig. 7

Thatte at Figure 7

Additionally, Thatte discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records in the linked list as garbage. Thatte at Col. 7, line 61- Col. 8, line 18.

accessing a linked list of records having same hash address,

Thatte in view of Dirks discloses accessing a linked list of records having the same hash address. Thatte at Col. 8, lines 39-62; Fig. 7. Specifically, Thatte discloses that the MMU in association with the reference count filter performs insert, delete, reconcile, and locate operations to access the linked list. Thatte at Col. 6, lines 64-68; Col. 8, line 55-62. As further disclosed in Thatte, a virtual address functions as a search key, as it is "applied to means for implementing a

hash function," wherein the "hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located" in a linked list, therefore having the same hash address. Thatte at Col. 8, lines 49-62; Fig. 7.

identifying at least some of the automatically expired ones of the records,

Thatte in view of Dirks discloses identifying at least some of the automatically expired ones of the records. Thatte at Col. 7, line 61- Col. 8, line 18; Col. 8, lines 57-59. In particular, Thatte discloses that the pointers to the records in the hash table of linked lists are either marked or unmarked by the MMU; the unmarked pointers indicate garbage records, and are removed, or evacuated, from the hash table of linked lists. Thatte at Col. 7, line 61- Col. 8, line 18.

removing at least some of the automatically expired records from the linked list when the linked list is accessed, and

Thatte in view of Dirks discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Thatte at Col. 7, lines 1-26; Col. 8, lines 57-59; Col. 7, line 60- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore expire and evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18.

inserting, retrieving or deleting one of the records from the system following the step of removing.

Thatte in view of Dirks discloses inserting, retrieving or deleting one of the records from the system following the step of removing. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18. Specifically, while performing the insertion operation, thereby accessing the linked list, Thatte discloses that the MMU will reconcile and therefore expire and evacuate the unmarked garbage records in the hash table of linked lists if the reference count filter is full. Thatte at Col. 7, lines 1-26; Col. 7, line 61- Col. 8, line 18. Thatte further discloses that after removing the expired records in the reconciliation operation, the insertion operation can be completed. Thatte at Col. 7, lines 19-26.

CLAIM 8

The method according to claim 7 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Thatte in view of Dirks discloses the method according to claim 7 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Thatte at Col. 6, lines 18-63; Col. 8, lines 39-62; Col. 7, line 61- Col. 8, line 18; Fig. 7; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Thatte discloses accessing a table of linked lists. Thatte at Col. 8, lines 59-62. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 8, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 8, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 8.

E. CLAIMS 1-8 ARE RENDERED OBVIOUS BY MORRISON IN VIEW OF THATTE UNDER 35 U.S.C. § 103

Please see the attached Exhibit CC-E, presenting claim charts for comparison of Morrison in view of Thatte with claims 1-8 of the '120 patent.

Reasons to combine:

One of ordinary skill in the art would have been motivated to combine the memory management unit in Thatte with the dynamic dictionary in Morrison in order to provide for efficiently determining the extent to which the automatically expiring records in Morrison should be removed. Both Morrison and Thatte are directed towards navigating a list-type data structure and removing inactive or expired records as efficiently as possible. Furthermore, both Morrison and Thatte disclose the use of a hashing technique in an effort to effectively allocate space within a memory system. Additionally, one of ordinary skill in the art would recognize that the result of combining Morrison with Thatte would be nothing more than the predictable use of prior art elements according to their established functions. The result would simply be the application of Thatte's reconcile operation to the linked list data structure disclosed in Morrison.

CLAIM 1

An information storage and retrieval system, the system comprising:

Morrison in view of Thatte discloses an information storage and retrieval system. Morrison at p. 1155, abstract. Specifically, Morrison discloses a dynamic dictionary that is used for records storage and retrieval. *Id.*

a linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring,

Morrison in view of Thatte discloses that records in a memory are inserted and sought in a separate-chaining hash table of linked lists. Morrison at p. 1155, abstract and ¶ 2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2 ("A sequence of items is given; each item includes a search key, a starting time, and an expiration time.").

a record search means utilizing a search key to access the linked list,

Morrison in view of Thatte discloses a record search means utilizing a search key to access the linked list. Morrison at p. 1155, ¶2. In particular, Morrison discloses that each record includes a search key, a starting time, and an expiration time. *Id.* Morrison further discloses that records stored in the table of linked lists are searched by hashing the search key. *Id.*

the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed, and

Morrison in view of Thatte discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed by hashing with lazy deletion. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses hashing with lazy deletion, wherein a record identifies a starting time and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.

Morrison in view of Thatte discloses means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

CLAIM 2

The information storage and retrieval system according to claim 1 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Morrison in view of Thatte discloses an information storage and retrieval system according to claim 1 further including means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 2, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

CLAIM 3

A method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring, the method comprising the steps of:

Morrison in view of Thatte discloses a method for storing and retrieving information records in a dynamic dictionary using a linked list to store and provide access to the records, at least some of the records automatically expiring. Morrison at p. 1155, abstract and ¶ 2. Morrison specifically discloses the use of a linked list that can be searched and accessed by hashing a search key. Morrison at p. 1155, ¶2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2

("A sequence of items is given; each item includes a search key, a starting time, and an expiration time.").

accessing the linked list of records,

Morrison in view of Thatte discloses accessing the linked list of records by hashing a search key. Morrison at p. 1155, abstract and ¶2.

identifying at least some of the automatically expired ones of the records, and

Morrison in view of Thatte discloses identifying at least some of the automatically expired ones of the records from the linked list. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that each record includes a search key, a starting time, and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed.

Morrison in view of Thatte discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

CLAIM 4

The method according to claim 3 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Morrison in view of Thatte discloses the method according to claim 3 further including the step of dynamically determining the maximum number of expired ones of the records to

remove when the linked list is accessed. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 4, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

CLAIM 5

An information storage and retrieval system, the system comprising:

Morrison in view of Thatte discloses an information storage and retrieval system. Morrison at p. 1155, abstract. Specifically, Morrison discloses a dynamic dictionary that is used for records storage and retrieval. *Id.*

a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring,

Morrison in view of Thatte discloses hashing with lazy deletion, which provides access to records stored in a memory of the system, the dynamic dictionary, and also discloses using a separate, or external, chaining technique to store the records with the same hash address, at least some of the records automatically expiring. Morrison at p. 1155, abstract and ¶2. In particular, the records disclosed in Morrison are "hashed by search key in a table of linked lists (separate

chains); each time an item is added to a list, any items of that list that the new item shows to be expired are deleted from that list." Morrison at p. 1155, ¶2.

a record search means utilizing a search key to access a linked list of records having the same hash address,

Morrison in view of Thatte discloses a record search means utilizing a search key to access a linked list of records having the same has address. Morrison at p. 1155, ¶2. In particular, Morrison discloses that each record includes a search key, a starting time, and an expiration time. *Id.* Morrison further discloses that records stored in the table of linked lists are searched by hashing the search key. *Id.* Morrison further discloses the case wherein hashing with lazy deletion uses a single bucket in the hash table, in which case the linked list of records would have the same hash address. Morrison at p. 1156, ¶3.

the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed, and

Morrison in view of Thatte discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed by hashing with lazy deletion. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses hashing with lazy deletion, wherein a record identifies a starting time and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.

Morrison in view of Thatte discloses means, utilizing record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records. Morrison at p. 1155, abstract and ¶2. More specifically, Morrison discloses using

hashing with lazy deletion to insert records, seek out and therefore retrieve records, and deleting records "that no longer need to be in the data structure." Morrison at p. 1155, abstract. Morrison further discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed; any records in the linked list that the new record shows to be expired are deleted from the list. Morrison at p. 1155, abstract and ¶2. Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id*.

CLAIM 6

The information storage and retrieval system according to claim 5 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Morrison in view of Thatte discloses the information storage and retrieval system according to claim 5 further including means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 6, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the

records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

CLAIM 7

A method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring, the method comprising the steps of:

Morrison in view of Thatte discloses a method for storing and retrieving information records in a dynamic dictionary using hashing with lazy deletion on a linked list to store and provide access to the records, and also discloses using a separate, or external, chaining technique to store records with the same hash address. Morrison at p. 1155, abstract and ¶ 2. Morrison specifically discloses the use of a linked list that can be searched and accessed by hashing a search key. Morrison at p. 1155, ¶2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2 ("A sequence of items is given; each item includes a search key, a starting time, and an expiration time."). Additionally, Morrison discloses the case wherein hashing with lazy deletion uses a single bucket in the hash table, in which case the linked list of records would have the same hash address. Morrison at p. 1156, ¶3.

accessing a linked list of records having same hash address,

Morrison in view of Thatte discloses accessing the linked list of records by hashing a search key. Morrison at p. 1155, abstract and ¶2. Morrison also discloses the case where hashing with lazy deletion uses only a single bucket in the hash table of linked lists, and therefore discloses a linked list of records having the same hash address. Morrison at p. 1156.

identifying at least some of the automatically expired ones of the records,

Morrison in view of Thatte discloses identifying at least some of the automatically expired ones of the records from the linked list. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that each record includes a search key, a starting time, and an

expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed, and

Morrison in view of Thatte discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

inserting, retrieving or deleting one of the records from the system following the step of removing.

Morrison in view of Thatte discloses inserting, retrieving, or deleting one of the records from the system following the step of removing. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that expired records are only deleted when they are encountered during an insertion operation, indicating that the insertion operation has not yet completed before the automatically expiring records are removed. Morrison at p. 1155, ¶2.

CLAIM 8

The method according to claim 7 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Morrison in view of Thatte discloses the method according to claim 7 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18. In particular, Thatte discloses that during an insertion operation, the MMU will determine whether the reference counter filter is full; if the reference counter filter is not full, then the MMU in Thatte dynamically determines that the maximum number of records to remove is zero.

Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7. If the reference counter filter is full, then the MMU in Thatte dynamically determines that the maximum number of records to delete is all of the garbage in the reference counter filter. *Id.* Under the broadest reasonable interpretation of claim 8, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. Therefore, under the broadest reasonable interpretation, Thatte certainly discloses dynamically determining whether to remove none of the records, as well as dynamically determining to remove some but not all of the records. Thatte at Col. 7, lines 21-26; Col. 7, line 40- Col. 8, line 18; Fig. 7.

F. CLAIMS 1-8 ARE RENDERED OBVIOUS BY DIRKS IN VIEW OF MORRIS UNDER 35 U.S.C. § 103

Please see the attached Exhibit CC-F presenting claim charts for comparison of Dirks in view of Morris with claims 1-8 of the '120 patent.

Reasons to combine:

One of ordinary skill in the art would have been motivated to combine the computer memory system disclosed in Dirks with the computer memory system disclosed in Morris to incorporate the linked list feature specifically disclosed in Morris into the memory system of Dirks. As both Dirks and Morris disclose systems and methods for allocating memory address controls using page table entries that are stored in hash tables, one of ordinary skill in the art would have understood how to combine the hashed page table with linked lists taught in Morris with Dirks. Moreover, one of ordinary skill in the art would recognize that the linked lists disclosed in Morris would improve similar systems and methods in the same way. Additionally, one of ordinary skill in the art would recognize that the result of combining Dirks with Morris would be nothing more than the predictable use of prior art elements according to their

established functions. The result would simply be Dirks' page table being implemented with a hashing function using linked lists/external chaining.

CLAIM 1

An information storage and retrieval system, the system comprising:

Dirks in view of Morris discloses an information storage and retrieval system. Dirks at Col. 1, lines 5-8; Col. 1, lines 33-50. Specifically, Dirks discloses a virtual memory management system that stores and retrieves mappings of virtual addresses to physical addresses in a page table. *Id.*; Dirks at Col. 2, line 66- Col. 3, line 31.

a linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring,

Dirks in view of Morris discloses a linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring. Specifically, Dirks discloses a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with the same hash address. For example, the virtual segment identifiers (VSIDs) are stored in the page table by use of a hashing function. Dirks at Col. 5, lines 10-31; Col. 9, lines 32-46. Furthermore, Morris discloses page table entries distributed by a hash function, such as those in Dirks, using linked lists or external chaining to resolve collisions. Morris at Col. 3, line 54 – Col. 4, line 24; Col. 6, lines 47-65. Additionally, Dirks discloses VSIDs that automatically expire. Specifically, "[a]t some point in time, the VSID becomes inactive." Dirks at Col. 6, lines 24-34.

a record search means utilizing a search key to access the linked list,

Dirks in view of Morris discloses a record search means utilizing a search key to access the linked list. Specifically, Dirks discloses hash table search means such as searching, creation, deletion, and sweeping. Dirks at Col. 5, lines 31-47; Col. 5, lines 66- Col. 6, line 15; Col. 5, lines 39-44; Col. 9, lines 39-47.

the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed, and

Dirks in view of Morris discloses that the record search means includes means for identifying and removing at least some expired records from the linked list when the linked list is accessed. Specifically, Dirks discloses that at some point VSIDs become inactive. Dirks at Col. 6, lines 24-30. These expired or inactive VSIDs are not removed "in one colossal step, for example after all of the free VSIDs have been allocated. Rather, the sweeping is carried out in an incremental, ongoing manner to avoid significant interruptions in the running of programs." Dirks at Col. 6, lines 39-44. Furthermore, "each time that a new range of addresses is allocated to a program, a limited number of entries in the page table are examined, to determine whether the addresses associated with those entries are no longer in use and the entries can be removed from the page table." *Id.*

means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.

Dirks in view of Morris discloses utilizing the record search means for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list. Specifically, at some point VSIDs become inactive. Dirks at Col. 6, lines 24-30. These expired or inactive VSIDs are not removed "in one colossal step, for example after all of the free VSIDs have been allocated. Rather, the sweeping is carried out in an incremental, ongoing manner to avoid significant interruptions in the running of programs." Dirks at Col. 6, lines 39-44. Furthermore, "each time that a new range of addresses is allocated to a program, a limited number of entries in the page table are examined, to determine whether the addresses associated with those entries are no longer in use and the entries can be removed from the page table." *Id.* Dirks similarly teaches that a limited number of entries can be examined each time a thread is deleted.

CLAIM 2

The information storage and retrieval system according to claim 1 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Dirks in view of Morris discloses the information storage and retrieval system according to claim 1, further including the means for dynamically determining the maximum number for

the record search means to remove in the accessed linked list of records. Morris at Col. 3, line 54 – Col. 4, line 24; Col. 6, lines 47-65; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Morris discloses accessing a table of linked lists. Morris at Col. 3, line 54 – Col. 4, line 24; Col. 6, lines 47-65. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 2, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 2, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 2.

CLAIM 3

A method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring, the method comprising the steps of:

Dirks in view of Morris discloses a method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring. Specifically, Dirks discloses a virtual memory management system that stores and retrieves mappings of virtual addresses to physical addresses in a page table. *Id.*; Dirks at Col. 2, line 66- Col. 3, line 31. Furthermore, Morris discloses page table entries

distributed by a hash function, such as those in Dirks, and then storing the page table entries in a hash table using linked lists or external chaining. Morris at Col. 3, line 54 – Col. 4, line 24; Figure 3. Additionally, Dirks discloses VSIDs that automatically expire. Specifically, "[a]t some point in time, the VSID becomes inactive." Dirks at Col. 6, lines 24-34.

accessing the linked list of records,

Dirks in view of Morris discloses accessing the linked list of records. Morris at Col. 3, lines 54-65. Specifically, "[f]or translating a given virtual address to a physical address, one approach is to perform a many-to-one function (hash) on the virtual address to form an index into the page table. This gives a pointer to a linked list of entries. These entries are then searched for a match." *Id.*

identifying at least some of the automatically expired ones of the records, and

Dirks in view of Morris discloses identifying at least some of the automatically expired ones of the records. Dirks at Col. 5, lines 31-47; Col. 5, line 66 – Col. 6, line 15; Col. 5, lines 39-44; Col. 9, lines 39-47. Specifically, Dirks discloses hash table search means such as searching, creation, deletion, and sweeping. *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed.

Dirks in view of Morris discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Specifically, at some point VSIDs become inactive. Dirks at Col. 6, lines 24-30. These expired or inactive VSIDs are not removed "in one colossal step, for example after all of the free VSIDs have been allocated. Rather, the sweeping is carried out in an incremental, ongoing manner to avoid significant interruptions in the running of programs." Dirks at Col. 6, lines 39-44. Furthermore, "each time that a new range of addresses is allocated to a program, a limited number of entries in the page table are examined, to determine whether the addresses associated with those entries are no longer in use and the entries can be removed from the page table." *Id.* Dirks similarly teaches that a limited number of entries can be examined each time a thread is deleted.

CLAIM 4

The method according to claim 3 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Dirks in view of Morris discloses the method according to claim 3 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Morris at Col. 3, lines 54-65; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Morris discloses accessing a table of linked lists. Morris at Col. 3, lines 54-65. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 4, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 4, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 4.

CLAIM 5

An information storage and retrieval system, the system comprising:

Dirks in view of Morris discloses an information storage and retrieval system. Dirks at Col. 1, lines 5-8; Col. 1, lines 33-50. Specifically, Dirks a virtual memory management system

that stores and retrieves mappings of virtual addresses to physical addresses in a page table. *Id.*; Dirks at Col. 2, line 66- Col. 3, line 31.

a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring,

Dirks in view of Morris discloses a hashing means to provide access to records stored in a memory of the system, and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring. Specifically, Dirks discloses that the virtual segment identifiers (VSIDs) are stored in the page table by use of a hashing function. Dirks at Col. 5, lines 10-31. "Through the use of the hashing function, the page table entries are efficiently distributed within the page table." Dirks at Col. 5, lines 25-27. Furthermore, Morris discloses external chaining to the extent that it provides for a method for searching page table entries (in order to find the physical address, given the virtual address in a reduced tag) wherein it sequentially searches through a linked list of page table entries by following the pointer in each page table entry to the next page table entry in the linked list. Morris at Col. 4, lines 1-15; Col. 5, lines 45-56. Additionally, Dirks discloses VSIDs that automatically expire. Specifically, "[a]t some point in time, the VSID becomes inactive." Dirks at Col. 6, lines 24-34.

a record search means utilizing a search key to access a linked list of records having the same hash address,

Dirks in view of Morris discloses a record search means utilizing a search key to access a linked list of records having the same hash address. In particular, Morris discloses accessing a table of linked lists. Morris at Col. 3, lines 54-65. Furthermore, Dirks discloses hash table search means such as searching, creation, deletion, and sweeping. Dirks at Col. 5, lines 31-47; Col. 5, lines 66- Col. 6, line 15; Col. 5, lines 39-44; Col. 9, lines 39-47.

the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed, and

Dirks in view of Morris discloses the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed. In particular, Morris discloses accessing a table of linked lists.

Morris at Col. 3, lines 54-65. Furthermore, Dirks discloses that at some point VSIDs become inactive. Dirks at Col. 6, lines 24-30. These expired or inactive VSIDs are not removed "in one colossal step, for example after all of the free VSIDs have been allocated. Rather, the sweeping is carried out in an incremental, ongoing manner to avoid significant interruptions in the running of programs." Dirks at Col. 6, lines 39-44. Furthermore, "each time that a new range of addresses is allocated to a program, a limited number of entries in the page table are examined, to determine whether the addresses associated with those entries are no longer in use and the entries can be removed from the page table." *Id.*

meals, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.

Dirks in view of Morris discloses utilizing the record search means for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records. In particular, Morris discloses accessing a table of linked lists. Morris at Col. 3, lines 54-65. Furthermore, Dirks discloses that at some point VSIDs become inactive. Dirks at Col. 6, lines 24-30. These expired or inactive VSIDs are not removed "in one colossal step, for example after all of the free VSIDs have been allocated. Rather, the sweeping is carried out in an incremental, ongoing manner to avoid significant interruptions in the running of programs." Dirks at Col. 6, lines 39-44. Furthermore, "each time that a new range of addresses is allocated to a program, a limited number of entries in the page table are examined, to determine whether the addresses associated with those entries are no longer in use and the entries can be removed from the page table." *Id.* Dirks similarly teaches that a limited number of entries can be examined each time a thread is deleted.

CLAIM 6

The information storage and retrieval system according to claim 5 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Dirks in view of Morris discloses the information storage and retrieval system according to claim 5, further including the means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Morris at Col. 3, lines 54-65; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Morris discloses accessing a table of linked lists. Morris at Col. 3, lines 54-65. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 6, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 6, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 6.

CLAIM 7

> **A method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring, the method comprising the steps of:**

Dirks in view of Morris discloses a method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with same hash address, at least some of the records automatically

expiring. Specifically, Dirks discloses a virtual memory management system that stores and retrieves mappings of virtual addresses to physical addresses in a page table. *Id.*; Dirks at Col. 2, line 66- Col. 3, line 31. Furthermore, Morris discloses external chaining to the extent that it provides for a method for searching page table entries (in order to find the physical address, given the virtual address in a reduced tag) wherein it sequentially searches through a linked list of page table entries by following the pointer in each page table entry to the next page table entry in the linked list. Morris at Col. 4, lines 1-15; Col. 5, lines 45-56. Additionally, Dirks discloses VSIDs that automatically expire. Specifically, "[a]t some point in time, the VSID becomes inactive." Dirks at Col. 6, lines 24-34.

accessing a linked list of records having same hash address,

Dirks in view of Morris discloses accessing a linked list of records having same hash address. In particular, Morris discloses accessing a table of linked lists. Morris at Col. 3, lines 54-65.

identifying at least some of the automatically expired ones of the records,

Dirks in view of Morris discloses identifying at least some expired ones of the records. Specifically, Dirks discloses that at some point VSIDs become inactive. Dirks at Col. 6, lines 24-30. These expired or inactive VSIDs are not removed "in one colossal step, for example after all of the free VSIDs have been allocated. Rather, the sweeping is carried out in an incremental, ongoing manner to avoid significant interruptions in the running of programs." Dirks at Col. 6, lines 39-44. Furthermore, "each time that a new range of addresses is allocated to a program, a limited number of entries in the page table are examined, to determine whether the addresses associated with those entries are no longer in use and the entries can be removed from the page table." *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed, and

Dirks in view of Morris discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. In particular, Morris discloses

accessing a table of linked lists. Morris at Col. 3, lines 54-65. Furthermore, Dirks discloses that at some point VSIDs become inactive. Dirks at Col. 6, lines 24-30. These expired or inactive VSIDs are not removed "in one colossal step, for example after all of the free VSIDs have been allocated. Rather, the sweeping is carried out in an incremental, ongoing manner to avoid significant interruptions in the running of programs." Dirks at Col. 6, lines 39-44. Furthermore, "each time that a new range of addresses is allocated to a program, a limited number of entries in the page table are examined, to determine whether the addresses associated with those entries are no longer in use and the entries can be removed from the page table." *Id.* Dirks similarly teaches that a limited number of entries can be examined each time a thread is deleted.

inserting, retrieving or deleting one of the records from the system following the step of removing.

Dirks in view of Morris discloses inserting, retrieving or deleting one of the records from the system following the step of removing. Specifically, Dirks discloses that during an insertion operation, the memory manager examines a number of record entries k in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses determining the maximum number of expiring records to be removed during the insertion operation k by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46.

CLAIM 8

The method according to claim 7 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Dirks in view of Morris discloses the method according to claim 7 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Morris at Col. 3, lines 54-65.; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Morris discloses accessing a table of linked lists. Morris at Col. 3, lines 54-65. Additionally, Dirks discloses that during an

insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 8, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 8, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 8.

VI. CONCLUSION

The prior art documents presented in the above request were either not previously considered by the PTO or are now being presented in a new light pursuant to MPEP § 2642(II)(A). Claims 1-8 of the '120 patent are not patentable over the prior art documents cited herein. The prior art documents teach the subject matter of the '120 patent in a manner such that substantial new questions of patentability for all claims are raised by this request.

In view of the foregoing, it is respectfully submitted that a substantial new question of patentability of claims 1-8 of U.S. Patent No. 5,893,120 has been raised by this request. Accordingly, the PTO is requested to grant this request and to initiate reexamination with special dispatch.

As an aid to the application of the presented prior art to claims of the '120 patent, corresponding claim charts are provided at Exhibits CC-A through CC-F attached hereto.

Enclosed is a credit card authorization to cover the Fee for reexamination. If this authorization is missing or defective please charge the Fee to the Novak Druce Deposit Account No. 14-1437.

Respectfully submitted,

/Lissi Mojica Marquis/

NOVAK DRUCE & QUIGG, LLP
1000 Louisiana St.
53rd Floor
Houston, Texas 77002
P: 713-571-3400
F: 713-456-2836

Novak Druce & Quigg, LLP
Donald J. Quigg
Reg. No. 16,030
Tracy W. Druce
Reg. No. 35,493
Lissi Mojica Marquis
Reg. No. 63,421