



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/514,674	10/15/2014	Calvin Hsia	333083.02	2271
69316	7590	09/01/2020	EXAMINER	
MICROSOFT CORPORATION ONE MICROSOFT WAY REDMOND, WA 98052			OTTO, ALAN	
			ART UNIT	PAPER NUMBER
			2132	
			NOTIFICATION DATE	DELIVERY MODE
			09/01/2020	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

chriochs@microsoft.com
usdoCKET@microsoft.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte CALVIN HSIA

Appeal 2019-000590
Application 14/514,674
Technology Center 2100

Before ST. JOHN COURTENAY III, MARC S. HOFF, and
KEVIN F. TURNER, *Administrative Patent Judges*.

HOFF, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellant¹ appeals under 35 U.S.C. § 134 from a final rejection of claims 1–5, 7–15, 17–19, and 21–23.² We have jurisdiction under 35 U.S.C. § 6(b).

We affirm-in-part.

Appellant's invention is analysis and presentation of allocated computer system memory. Spec. ¶ 10. An analysis component can be configured to analyze the computer memory in connection with data

¹ Appellant states that the real party in interest is Microsoft Technology Licensing, LLC. Appeal Br. 1.

² Claims 6, 16, and 20 have been cancelled.

included in at least one tag based upon input received. Spec. ¶ 11. Inefficient use of computer memory can be identified, and an interface component can be configured to present output generated in connection with the inefficient use of computer memory. Spec. ¶ 15. Memory allocation can be tracked from a process that issues the memory allocation or from a garbage collection process. Spec. ¶ 39.

Claim 1 is reproduced below:

1. A method performed by a computing system having a processor and memory, the method comprising:

detecting a memory allocation instruction to allocate a portion of the memory, the memory

allocation instruction being issued by a process executed by the processor;

in response to the detected memory allocation instruction for allocating a portion of the memory,

allocating a portion of the memory;

creating a tag corresponding to the allocated portion of the memory, the tag including information of a call stack associated with the monitored memory allocation instruction; and

storing the created tag in a private heap of the memory; subsequently,

detecting a memory release instruction to release the allocated portion of the memory;

in response to the detected memory release instruction, deleting the stored tag corresponding to the allocated portion of the memory from the private heap of the memory; and

tracking the allocated portion of the memory during a garbage collection process and linking the information of the call stack that is associated with the monitored memory allocation instruction and is contained in the created tag with another portion of the memory instead of the allocated portion

of the memory when data in the allocated portion of the memory is moved to the another portion of the memory during the garbage collection process.

The prior art relied upon by the Examiner as evidence is:

Name	Reference	Date
Gold	US 2002/0116573 A1	Aug. 22, 2002
Krapp	US 6,889,297 B2	May 3, 2005
Kurtz	US 2007/0011428 A1	Jan. 11, 2007
Patterson	US 8,028,009 B1	Sept. 27, 2011

Claims 1–5, 7, 11–15, and 17 stand rejected under 35 U.S.C. § 103 as being unpatentable over Kurtz and Patterson. Final Act. 2.

Claims 8, 18, 19, and 21–23 stand rejected under 35 U.S.C. § 103 as being unpatentable over Kurtz, Patterson, and Krapp. Final Act. 6.

Claims 9 and 10 stand rejected under 35 U.S.C. § 103 as being unpatentable over Kurtz, Patterson, Gold, and Krapp. Final Act. 18.

Throughout this decision, we make reference to the Appeal Brief (“Appeal Br.,” filed May 21, 2018), the Reply Brief (“Reply Br.,” filed Oct. 30, 2018) and the Examiner’s Answer (“Ans.,” mailed Aug. 31, 2018) for their respective details.

ISSUES

1. Does the combination of Kurtz and Patterson fairly suggest tracking the allocated portion of the memory during a garbage collection process?

2. Does Kurtz teach native code and detecting the memory allocation instruction from the native code?

3. Does the combination of Kurtz and Patterson suggest detecting the memory release instruction from a garbage collection process?

4. Does the combination of Kurtz and Patterson suggest tagging an objection associated with the allocated portion of the memory with a number

of times the object has survived or moved during garbage collection process?

PRINCIPLES OF LAW

One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *In re Keller*, 642 F.2d 413, 425 (CCPA 1981). The test of obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference, nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. *Id.* at 425.

Claim construction is an important step in a patentability determination. “Both anticipation under § 102 and obviousness under § 103 are two-step inquiries. The first step in both analyses is a proper construction of the claims. . . . The second step in the analyses requires a comparison of the properly construed claim to the prior art.” *See Medichem, S.A. v. Rolabo, S.L.*, 353 F.3d 928, 933 (Fed. Cir. 2003) (internal citations omitted)). Under the second step, the Board must compare the construed claim to one or more prior art references and make factual findings regarding the limitations contested by Appellants. *See In re Crish*, 393 F.3d 1253, 1256 (Fed. Cir. 2004).

ANALYSIS

Claims 1–3 and 11–13

Independent claims 1 and 12 recite, in pertinent part, “tracking the allocated portion of the memory during a garbage collection process and

linking that information of the call stack that is associated with the monitored memory allocation instruction . . . with another portion of the memory.”

The Examiner rejects the claims over Kurtz and Patterson. The Examiner finds that Kurtz does not teach this “tracking” limitation in full, but finds that Patterson teaches such tracking of an allocated portion of memory during a garbage collection process. Final Act. 3–4. Appellant argues in the Brief that Patterson does not teach this tracking limitation, but rather “teaches tracking blocks of data contained in certain memory locations during a garbage collection process.” *Id.*

In response to Appellant, the Examiner finds that “Kurtz already teaches tracking allocated portions of memory. Kurtz is being combined with Patterson which teaches where memory tracking is done during a garbage collection process.” Ans. 4; Kurtz ¶¶ 44, 45. As Kurtz allocates memory, Kurtz constructs a memory tag 200, populated with pertinent information (activity type, allocation address and size, task identification, stack depth, traceback stack, previous flag, next flag), that is “stored on the memory tag list along with other memory tags 200 that may currently exist.” Kurtz ¶¶ 46, 47. When memory is de-allocated, the corresponding memory tag is removed from the memory tag list. Kurtz ¶¶ 49. Appellant admits that Kurtz uses memory tag 200 to track memory allocations. Appeal Br. 10.

Appellant argues that Patterson, relied upon for disclosing tracking allocated portions of memory during garbage collection processes, actually teaches “tracking blocks of data contained in certain memory locations.” Appeal Br. 9. Patterson discloses that a block of data can be identified by a block ID, and a memory location in which the block of data is stored can be

identified in the location table. *Id.*; Patterson, Fig. 3, col. 9:1–3. When a block of data is moved, only the address column in the location table needs to be updated. *Id.*; Patterson, col. 9:49–54. We do not agree with Appellant’s argument. We agree with the Examiner’s proposed modification of Kurtz’s tracking of allocated portions of memory with Patterson’s tracking during garbage collection processes, “in order to only update one location in the table when blocks are moved and copied.” Final Act. 5. As the Examiner points out, Patterson’s location table shows a block ID and an address of the data, which corresponds to an allocated portion of memory. Ans. 4; Patterson, Fig. 3. We agree with the Examiner that Appellant’s claims do not specify the range of memory that is allocated, or the number of blocks in the range; thus, Appellant’s argument that Patterson’s data block may not correspond to the size of the memory allocation is not persuasive of Examiner error. Ans. 4.

Appellant argues that neither Kurtz nor Patterson teach updating Kurtz’s memory tag 200 containing the call stack and the corresponding allocated memory location during a garbage collection operation. Appeal Br. 10. We are not persuaded by Appellant’s argument. Appellant here argues against the bodily incorporation of Patterson into Kurtz, but the test of obviousness is what the combined teachings of the references would have suggested to the person having ordinary skill in the art. *In re Keller*, 642 F.2d at 425. We agree with the Examiner that Kurtz teaches tracking allocated portions of memory, and that Patterson teaches updating tracking during garbage collection operations. Ans. 4.

Appellant’s argument that combining Kurtz and Patterson would change Kurtz’s principle of operation is similarly unpersuasive. Appeal Br.

11. As noted *supra*, we agree with the Examiner’s finding that Kurtz teaches tracking allocated portions of memory, and that Patterson’s tracking includes a block ID and an address of the data.

Appellant’s arguments are not persuasive to show that the Examiner erred in rejecting the claims over Kurtz and Patterson. We sustain the Examiner’s § 103 rejection of claims 1–3 and 11–13.

Claims 4 and 14

Appellant argues that Kurtz teaches that the memory tool 70 allocates memory from the memory pool 90 to the applications 80 and audits the usage of such memory. Appeal Br. 12. Assuming that the memory tool corresponds to claim 4’s “managed code,” Appellant urges, Kurtz would teach detecting both the memory allocation instruction and the memory release instruction from managed code. *Id.*

The Examiner finds that Kurtz teaches a memory allocation instruction, “malloc,” in native code (C). Ans. 5; Kurtz, Fig. 6. As a result, the Examiner then finds that Kurtz teaches a process including native code, as claimed, and detecting the memory allocation instruction from the native code, as claimed.

We do not agree with the Examiner. In our review of the record, we find that the Examiner has not done a proper construction of the claim term “native code,” which is a prerequisite to comparing a properly construed claim to the prior art. *Medichem*, 353 F.3d at 933. On this record, we cannot agree with the Examiner that the “C” language memory allocation instruction “malloc” corresponds to “native code,” which generally corresponds to the native instruction set of a particular microprocessor, as is known in the art. Thus, we find that the Examiner erred in determining that

Kurtz and Patterson teach or suggest all the elements of claims 4 and 14. We do not sustain the Examiner's § 103 rejection of these claims.

Claims 5 and 15

Appellant argues that Patterson describes how a mark and sweep process operates, not “detecting the memory release instruction from a garbage collection process” as claimed. Appeal Br. 12–13.

We are not persuaded by Appellant's argument. We agree with the Examiner that Patterson teaches that after data are copied to a different location during a sweep, “[t]he range of the address space of the data that has been swept is then marked as unallocated, thereby allowing the backup system to overwrite this range of the address space with new data.” Patterson, col. 1:53–56. We agree with the Examiner that Patterson teaches detecting the memory release instruction from a garbage collection process, as claimed. We sustain the Examiner's § 103 rejection of claims 5 and 15 over Kurtz and Patterson.

Claims 7 and 17

Claim 7 recites, in pertinent part, “tagging an objection associated with the allocated portion of the memory with a number of times the object has survived or moved during garbage collection processes.” The Examiner cites to Kurtz as teaching “where any event can be counted or tracked. The event of moving a memory object would be a memory event.” Final Act. 6; Kurtz ¶¶ 40, 49. Claim 17, dependent from claim 12, recites similar limitations.

Kurtz teaches a portion of a listing derived from a memory tag, with portion 310 representing “the number of times a similar event occurred.” Kurtz ¶ 40. Kurtz further teaches, during memory de-allocation, that in

removing a memory tag 200 from the memory tag list, a variety of counters may still be maintained, “for example, to account for the number of times certain events have occurred.” Kurtz ¶ 49.

We do not agree with the Examiner that these general disclosures of counting “the number of times certain events have occurred” may fairly be characterized as teaching the claimed “tagging an objection associated with the allocated portion of the memory with a number of times the object has survived or moved during garbage collection processes.”

Appellant has demonstrated that the Examiner erred in rejecting claims 7 and 17. We do not sustain the Examiner’s § 103 rejection.

Rejection of claims 8, 18, 19, and 21–23

With respect to independent claim 19, Appellant argues only that Krapp does not cure the deficiencies of Kurtz and Patterson. Appeal Br. 16. Since, as discussed *supra*, we conclude that the Examiner did not err in rejecting claims 1 and 12, we sustain the Examiner’s rejection of claim 19 over Kurtz, Patterson, and Krapp, for the same reasons given with respect to the rejection of claims 1 and 12 over Kurtz and Patterson, *supra*.

Appellant does not argue dependent claims 8, 18, and 21–23 separately, remarking only that these claims are allowable for the reasons given with respect to parent claims 1 and 12. Appeal Br. 16. We therefore sustain the rejection of claims 8, 18, and 21–23 over Kurtz, Patterson, and Krapp, for the same reasons given with respect to the rejection of claims 1 and 12 over Kurtz and Patterson, *supra*.

Rejection of claims 9 and 10

Appellant does not respond to the § 103 rejection of claims 9 and 10 over Kurtz, Patterson, Gold, and Krapp in the principal Appeal Brief or in

the Reply Brief. Accordingly, we sustain *pro forma* the Examiner's § 103 rejection of claims 9 and 10.

CONCLUSIONS

1. The combination of Kurtz and Patterson fairly suggests tracking the allocated portion of the memory during a garbage collection process.

2. Kurtz does not teach native code and detecting the memory allocation instruction from the native code.

3. The combination of Kurtz and Patterson suggests detecting the memory release instruction from a garbage collection process.

4. The combination of Kurtz and Patterson does not suggest tagging an objection associated with the allocated portion of the memory with a number of times the object has survived or moved during garbage collection process.

DECISION SUMMARY

In summary:

Claims Rejected	35 U.S.C. §	Reference(s)/ Basis	Affirmed	Reversed
1-5, 7, 11-15, 17	103	Kurtz, Patterson	1-3, 5, 11-13, 15	4, 7, 14, 17
8, 18, 19, 21-23	103	Kurtz, Patterson, Krapp	8, 18, 19, 21-23	
9, 10	103	Kurtz, Patterson, Gold, Krapp	9, 10	
Overall Outcome			1-5, 8-15, 18, 19, 21-23	4, 7, 14, 17

Appeal 2019-000590
Application 14/514,674

ORDER

The Examiner's decision to reject claims 1-3, 5, 8-13, 15, 18, 19, and 21-23 is affirmed. The Examiner's decision to reject claims 4, 7, 14, and 17 is reversed.

TIME PERIOD FOR RESPONSE

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a). *See* 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED-IN-PART