THIS OPINION WAS NOT WRITTEN FOR PUBLICATION

The opinion in support of the decision being entered today (1) was not written for publication in a law journal and (2) is not binding precedent of the Board.

Paper No. 13

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte JOHN J. PEARCE

Appeal No. 1998-1104
Application 08/380,985

ON BRIEF

Before HAIRSTON, FLEMING, and FRAHM, Administrative Patent Judges.

HAIRSTON, Administrative Patent Judge.

DECISION ON APPEAL

This is an appeal from the final rejection of claims 1-11.

The disclosed invention relates to a method and apparatus for optimizing the location in system memory of machine-

dependent code for implementing a plurality of functions comprising an applications program interface (API). The system memory is comprised of secure memory space and nonsecure memory space.

Claim 1 is illustrative of the claimed invention, and reads as follows:

1. In a computer system comprising system memory and a processor capable of operating in a system management mode (SMM), a method for enabling the optimization of the location in said system memory of machine-dependent code for implementing a plurality of functions comprising an applications program interface (API), a portion of said system memory comprising secure memory space accessible only while said SMM is invoked, the method comprising:

storing code for implementing less than all of said plurality of functions in said secure memory space;

storing code for implementing each of the remaining ones of said plurality of functions in memory space other than said secure memory space;

responsive to an applications program calling one of said plurality of functions, determining whether code for implementing said called one of said plurality of functions is stored in said secure memory space;

responsive to a determination that said code for implementing said called one of said plurality of functions is stored in said secure memory space;

invoking said SMM;

executing said code for implementing said called one of said plurality of functions; and

returning from said SMM; and

responsive to a determination that said code for implementing said called one of said plurality of functions is not stored in said secure memory space, executing said code for implementing said called one of said plurality of functions.

The examiner relies on the following references:

Dayan et al. (Dayan) 5,063,496 Nov. 5, 1991

Thorson, M. (Thorson), "System management mode explained; despite common functions, implementation details differ," Microprocessor Report, Vol. 6, No. 8, page 14(4), June 17, 1992.

In addition, the examiner also relied upon admitted prior art set forth on pages 2 and 3 of the specification.

Claims 1-11 stand rejected under 35 U.S.C. § 103 as being unpatentable over Dayan in view of Thorson, and in further view of the admitted prior art.

Reference is made to the brief and the answer for the respective positions of the appellant and the examiner.

OPINION

After careful consideration of the record before us, we will not sustain the 35 U.S.C. § 103 rejection of claims 1-11.

According to the examiner (Answer, pages 4-5), Dayan teaches the claimed apparatus and method steps but "does not teach operating in the system management mode and an applications program interface (API)." For such teachings, the examiner turns to Thorson (Answer, page 5). Based upon the teachings of Thorson, the examiner contends (Answer, page 5) that "[i]t would have been obvious to a person of ordinary skill in the art at the time the invention was made to operate Dayan et al's system in SMM and include API functions to service the application programs because SMM is designed to operate in multitasking environments like that of Dayan's which utilize both protected and real modes to efficiently negotiate each mode of operation and APIs provide reliable interfaces to an application."

Appellant argues (Brief, pages 4-5) the following:

> In particular, as discussed in the "Background of the Invention," one of the principal benefits of SMM is that it provides a secure memory location for status and control code; however, the price of this security is a significant reduction in efficiency. In some platform designs, extended applications program interfaces (APIs), such as Advanced Power Management (APM), Plug and Play (PnP) and other machine dependent programs, need to run with maximum efficiency with respect to

part of their functionality, but with maximum security with respect to other parts. Typically, this problem is solved by sacrificing efficiency for security and executing the entire extended API in the SMM.

The claimed invention provides a solution to this problem[.][sic, by] Applicant has solved this problem by storing code for implementing some (i.e., less than all) of a plurality of the functions of an API in secure (i.e., SMM) memory space and storing code for implementing the rest of the functions of the same API in non-secure memory space, a feature which is neither taught nor suggested by the cited combination of references. In addition, because none of the cited references teach or suggest storing code for implementing certain functions in secure memory space and code for implementing the remaining functions in unsecure memory space, it logically follows that the references also fail to teach determining whether code for implementing a called one of the API functions is stored in said secure memory space, as all functions are stored in the same memory space.

As conceded by the Examiner, Dayan fails to teach operating in [a] system management mode and an applications program interface, for which Thorson and Admitted Prior Art are respectively cited. However, even assuming *arguendo* that the references are properly combinable, the combination teaches, at best, executing the entire API in SMM, which is exactly the problem solved by Applicant's invention (see [Specification,] page 3, lines 4-11).

We agree with appellant's arguments concerning the teachings, and the lack of teachings, in both Dayan and Thorson. The Thorson reference is nothing more than the admitted prior art from the Specification (pages 2-3). Thorson discloses that SMM is implemented in conjunction with the Intel™ 386 SL CPU (Specification, pages 2-3). In Thorson, it appears that all of the API functions are performed in the SMM environment as opposed to being done between a SMM and a non-SMM environment as claimed by appellant supra. A capsulized version of the teachings of Dayan is taken from the ABSTRACT as follows:

> The BIOS routines [22,30] are accessed through protected entry points [28]. When an application program [24] attempts to access one of the routines by using a hard coded instruction for jumping to such entry point, a BIOS signaling routine [26] is executed which provides a signal to an operating system [20] allowing the operating system to control the access without being bypassed.

In short, the teachings of Dayan are not relevant to the claims on appeal which are directed to SMM and non-SMM for performing functions of an API.

As indicated supra, in the admitted prior art, all of the API functions are performed in the secure environment of the SMM. Thus, Dayan in combination with Thorson and the admitted prior art does not disclose API functions performed between a SMM and a non-SMM environment.

In summary, the rejection of claims 1-11 is reversed because a prima facie case of obviousness has not been made by the examiner.

DECISION

The decision of the examiner rejecting claims 1-11 under 35 U.S.C. § 103 is reversed.

REVERSED

Kenneth W. Hairston Administrative Patent Judge	)	
Michael R. Fleming Administrative Patent Judge	)	BOARD OF PATENT APPEALS AND INTERFERENCES
Eric Frahm Administrative Patent Judge	)	

tdl

David L. McCombs
HAYNES AND BOONE, L.L.P.
3100 NationsBank Plaza
901 Main Street
Dallas, TX 75202-3789