Software Interrupt Control Module Proposal PE-T-1004, Rev. 1

September 22, 1982 DATE:

TO: R and D Personnel

Jerry Kazin FROM:

SUBJECT: Software Interrupt Control Module Proposal

REFERENCE: PRIMOS Subroutines Guide (BREAK\$ Function) Software Interrupt Mechanism PE-TI-879 Software Interrupt Control Module Functional Spec. PE-T-1005 Software Interrupt Control Module Design Spec. PE-TI-1006

KEYWORDS: Software Interrupts, QUIT

ABSTRACT

Prior to Rev. 18, PRIMOS contained only one software interrupt known as the QUIT\$ condition. This condition may be enabled/disabled by calling the BREAK\$ module. At both Rev. 18 and Rev. 19 new software interrupts were added to the system. Unfortunately, no method has yet been provided for the users to enable/disable these new interrupts. This is the problem addressed by this proposal.

> This document is classified PRIME RESTRICTED. It must not be distributed to non-Prime Personnel. When there is no longer a need for this document, it should be returned to the Bldg. 10 Information Center by special delivery inter-office mail - or destroyed.

> > ©Prime Computer, Inc., 1983 All Rights Reserved

> > > *** PRIME RESTRICTED ***

Software Interrupt Control Module Proposal PE-T-1004, Rev. 1

s² 44

Table of Contents

1	BACKGROUND
2	PROPOSAL
3	HOW SW\$INT WILL AFFECT SYSTEM PROGRAMMERS
4	SOLUTION SUMMARY
5	MORE INFORMATION ON THE SOFTWARE INTERRUPT MECHANISM

-3 - ...

Software Interrupt Control Module Proposal PE-T-1004, Rev. 1

1 BACKGROUND

Prior to Rev. 18, PRIMOS contained only one software interrupt known as the QUIT\$ condition. This condition may be enabled/disabled by calling the BREAK\$ module. For more infromation on the BREAK\$ module see PRIMOS Subroutines Guide (BREAK\$ Function).

At Rev. 18 the following two interrupts with their respective conditions were added:

1) CPU watchdog timer (CPU TIMER\$ condition) and

2) Real Time watchdog timer (ALARM\$ condition).

At Rev. 19 the following three interrupts with their respective conditions were added:

1) Phantom Logout Notification (PH LOGO\$),

2) Cross Process Signalling (CPS\$), and

3) Logout (LOGOUT\$).

CPS\$ is an internal condition and will never be released to the general public. Unfortunately, no method has yet been provided for the users to enable/disable these new interrupts in either Rev. 18 or Rev. 19.

Moreover, the existing BREAK\$ functionality, as an undocumented by-product, when called to disable QUITS, disables all software interrupts. This is not acceptable as some users, specifically PR1ME subsystems such as DBMS, cannot have all software interrupts turned off by BREAK\$.

Unfortunately, as Rev. 18 has been released, this functionality is out in the field. Fortunately, since it is not documented, no user may validly depend on this by-product of BREAK\$. In other words, to the user, no method has yet been provided to turn off software interrupts except for QUITS, and to PRIMOS, we do not have to support this BREAK\$ effect.

2 PROPOSAL

To allow users to control the enable/disable setting of the various software interrupts, a new module, SW\$INT, will be built. Its calling sequence is

call sw\$int(key, selection, value, ercode, [outer_ring]);

key - argument which specifies to either turn interrupt(s) on, turn interrupt(s) off, or read the present value of

*** PRIME RESTRICTED ***

Software Interrupt Control Module Proposal

PE-T-1004, Rev. 1

42 1 × 10

interrupt(s)

- value the returned value of the selected interrupt(s) when the read key has been passed
- ercode standard error code
- outer_ring an optional outer ring which when included specifies to sw\$int what ring to work in

The selection and value arguments are varying length bit strings. Each bit will be associated with one specific interupt type. The placement of each bit within the string does not have to be one to one with the internal placement of the bit as used by the PRIMOS software interrupt handler. This makes the SW\$INT bit definition(s) independent of the physical bit definition(s) and makes the internal representation transparent to the caller.

Normally SW\$INT turns on/off and/or reads the status of the interrupt(s) chosen in the ring of its caller. Including the outer ring argument makes SW\$INT operate on the software interrupt control in the passed ring.

Presently, there are no plans to release SW\$INT to the general public. It will be released to PR1ME subsystems for their use. This will allow us to release to the general public, at a future time, a gate which does not know of all the software interrupt types. Some subsystems, such as DBMS, cannot ever afford to have certain interrupt types, such as logout, turned off.

When a user turns on an interrupt which had been seen while disabled, a signal will be immeadiately be generated for that interrupt condition.

Note that software interrupts normally start out enabled in the outer rings and disabled in ring 0.

3 HOW SW\$INT WILL AFFECT SYSTEM PROGRAMMERS

The following sections describe the effects of SW\$INT's introduction as it will be perceived by system programmers and who these programmers will be. Software Interrupt Control Module Proposal

3.1 What Will Rev. 18 Users See?

- F - A

SW\$INT will not be integrated into Rev. 18. This means that PRIMOS will never provide a mechanism for turning on/off the CPU and Real Time Watchdogs at this Rev. This is exactly what we have today so it seems justified. Again, any user that calls BREAK\$ to turn off any interrupt other than QUIT is making use of an undocumented feature and does not have to be supported.

Note that this means that no changes will have to go into Rev. 18.

3.2 What Will Rev. 19 Users See?

SW\$INT will be the only module that can turn on/off software interrupts at Rev. 19 with the exception of BREAK\$ which will only affect QUITS. Either SW\$INT or BREAK\$ may be used to turn on/off QUITS.

3.3 What Do System Programmers Need To Know?

Basically, system programmers only need to know about two different areas to which SW\$INT relates. The first is the three classes of software interrupts that are found in PRIMOS today. The second is the way in which software interrupts are handled in ring 0.

3.3.1 Software Interrupt Classes

The easiest type to explain is the simple on/off interrupt. The CPU watchdog timer is one of these. These interrupts take on only an on or off state. There is no chance that multiple instances of these interupts will occur.

The second form of software interrupt is the counted interrupt such as QUITS. A counter is used to determine how many times at a given level the interrupt has been turned on/off. One may think of the counter as a stack. SW\$INT will not maintain these counters. Separate modules/mechanism which SW\$INT calls will do so. BREAK\$ does this for QUITS.

The third type of interrupts is the queued data interrupt. Phantom logout notification is one of these. The queuing mechanisms for these interrupt types will be responsible for queueing the data. SW\$INT will be responsible for turning the interrupt on/off.

3.3.2 Software Interrupts In Ring O

In ring 0 the sense of the normal state of software interrupts is reversed from the state in the outer rings. In the outer rings, interrupts are normally enabled. In ring 0, interrupts are normally disabled. This allows ring 0 programmers to forget about the need for creating critical sections in ring 0. Ring 0 is normally a critical

*** PRIME RESTRICTED ***

Software Interrupt Control Module Proposal PE-T-1004, Rev. 1

1 . Carl - Star

section. Additionally, the interrupt handling code in ring 0 when it sees a software interrupt and the execution is in ring 0 defers the taking of the interrupt until execution leaves the inner ring. In this manner, PRIMOS never loses software interrupts. For more information about this defering of interrupts see Software Interrupt Mechanism PE-TI-879.

3.4 Who Will Be SW\$INT's Initial Users

Both the Data Base Management and the TP groups have requirements for the watchdog timers. They will make use of SW\$INT. Phantom logout notification already has a control module which turns on/off notification. It will be superceded by SW\$INT.

4 SOLUTION SUMMARY

A module, SW\$INT will be introduced into Rev. 19 which will allow users to enable/inhibit specific software interrupt types. This module will be bit string oriented to make it easy for programmer's use. No changes will be made to Rev. 18. Finally, SW\$INT will be built in a manner which will make the interrupt mechanism(s) transparent to the user and will allow the easy addition of new interrupt types or classes.

15 MORE INFORMATION ON THE SOFTWARE INTERRUPT MECHANISM

For information on the details of the software interrupt mechanism consult Software Interrupt Mechanism PE-TI-879.

For information relating to the use of the new software interrupt control module consult Software Interrupt Control Module Functional Spec. PE-T-1005.

For information relating to the design details of the software interrupt control mechanism consult Software Interrupt Control Module |Design Spec. PE-TI-1006.

Page 6