Protection in EMAS 2900

[This document is in the form of an addendum to Reference 1.]

2900 Architecture provides a variety of methods to enable an operating system to maintain its own integrity.

- **Privilege**. A bit in the PSW is required to execute privileged instructions.
- **Activate**. A privileged instruction to start executing a less trusted program at a lower ACR level and on a new stack.
- **Out**. An instruction exactly comparable to the IBM **SVC**. Enables an untrusted program to return to Supervisor in a controlled and secure way.
- **ACR levels**. Four bits in the **PSW** define the *Access Control* status of the running program. All store accesses (except to the current stack segment) are preceded by a check of the current ACR level against the read or write permission in the segment table. An untrusted program can not read or damage system data.
- **System calls**. Procedure calls normally go via a descriptor but leave the stack and ACR unchanged. If a system call descriptor is supplied, the caller's ACR and PRIV are validated via a table and the call continues at a new ACR and PRIV. Calls may be made up or down the privilege hierarchy and may specify the same or a new (empty) stack, in which case firmware copies the parameters.

EMAS 2900 hoped to perform all its protection using the 15 ACR levels and the System call - this proved impossible since:-

- The stack is unprotected, so calls down the hierarchy must specify new stacks; although a return will set the stack back a system call does not and this leaves Supervisor without its global variables.
- The system call does not check stack space - it is possible to construct a program that system calls a privileged procedure leaving it exactly 0 bytes of stack - which is catastrophic! This does not normally happen since compilers require a margin of 4096 stack bytes before compiling a call.

The scheme implemented is to use System Call between User at ACR 10 and Director where failure is user-unfriendly but does not impair system integrity. The ACR checks allow system processes at ACR 5 to have access to the complete Director interface while user processes can only access a subset; this latter feature is very hard to reproduce on IBM architecture.

Director at ACR 2 uses **Out** to enter its Local Controller in a secure manner. **Local Controller** exits via the **Activate** instruction.

Local Controller and **Global Controller** both run at ACR1; there is one Global Controller stack which forks if there is more than one CPU. Each Local Controller incarnation has its own stack. Transitions between Local and Global are made by a machine code sequence that is effectively a PSW swop. This enables the Global Controller tables to appear as Global variables to each and every Local Controller. Since Local Controller is compiled as a subroutine of Global Controller this is very elegant and consistent.

Supervisor and **Director** tables are normally set with the write permission equal to the owner's ACR and read permission of 5. Tables may thus be read by privileged users and system processes but not by normal users. One data segment may be read at all ACR levels and contains the time of day and other public information.

Retrospect on Yesterday's Design

When EMAS 2900 was designed it was not realised that firmware assistance for **System Calls** would not be available on the smaller models, e.g. 2946 and 2950. If firmware is missing (or present but enabling checks fail) a jump is made to a nominated PC. Recovery from this is difficult but it is possible to perform the System Call by software. Since any System Call can be made to fail by specifying ridiculous ACRs it would have been possible to do all System Calls by software and provide calls onto existing stacks as well as inserting the missing checks on stack space. The original ring based scheme could be implemented; however, soft calls cost about 2000 instructions - a formidable overhead.

Conclusion

IBM-type protection consisting of 2 states and an SVC to communicate between them is simple and well understood. Rings are superior if they are well thought out and free from logical flaws. Regrettably 2900 has a deficiency in its ACR scheme and the Kernel of EMAS has to protect itself with the old rather than the new technology.

P.D. Stephens

Reference

1. D.J.Rees and P.D.Stephens: "The Kernel of the EMAS 2900 Operating System", Software - Practice and Experience, 12, 655-667 (1982).

This document was converted from plain text (with Scribe markup) to Markdown, HTML and Word format by Bob Eager, October 2016.