SAP Memory Management Implementations (In Process)

Purpose

Help you to understand the different SAP memory management implementations available in the SAP system.

Overview

This page contains information about the SAP memory management implementations and their differences, demonstrating the relevant parameters for each one of them.

SAP Memory Management Implementations

The parameter es/implementation is responsible for controlling which memory implementation is active in the SAP system. Depending on your operating system, different implementation strategies can be used, which consequently brings different parameters to control the Extended Memory area. The parameter ES/TABLE also modifies the behavior of the Extended Memory implementation. This parameter needs to be set to a different value depending on the Operating System.

Memory Implementations "map" and "view"

The memory implementations "map" and "view" represent the same strategy, however, as highlighted in the above image, the "map" implementation is used in Linux and Unix systems (except AIX), while the "view" implementation is the default for the Windows operating system.

Strategy and Characteristics

The "map" memory implementation causes the activation of a significantly simplified method for configuring memory management. This implementation is recommended for 32-bit Linux operating systems as described in detail in SAP Note 386605.

In this memory implementation strategy, only the currently active user context is mapped to the address space of the work process. This is considered a classic model where a roll-in corresponds to mapping a context in the extended memory, and a roll-out corresponds to unmapping this context. The below image shows an example of this process.
As represented above, each work process has an extended memory area reserved to the mapping of the current user context. The context is mapped from the shared resource, which contains all the user contexts, and once the processing is over, the context is returned back in the shared resource by making the context visible again.

**Parameters and Definitions**

- **em/initial_size_MB**: This parameter specifies how much SAP extended memory is available on the SAP system.
- **em/max_size_MB**: Maximum size of the SAP extended memory.
- **em/address_space_MB**: Quantity of Extended Memory that each single user context may fully occupy provided ztta/roll_extension has not yet been reached; this interval is reserved from the start of the instance.

**Memory Implementations "STD" and "flat"**

The memory implementations "STD" and "flat" represent the same strategy, however, as highlighted in the above image, the "STD" implementation is used in Linux and Unix systems, while "flat" is valid only for Windows.

**Strategy and Characteristics**

In this memory implementation strategy, the extended memory uses a memory-mapped file with memory protection. All the user contexts are always mapped in the virtual address space for each work process in the SAP system. All the user contexts are initially protected.

The roll-in, in this memory implementation strategy, means removing the protection from a user context in the extended memory and roll-out means protecting a context again. The advantage of this memory model is the reduced number of soft page faults, because the process working set is stable. The disadvantage is the overhead of the memory protection, which results in higher kernel-mode CPU usage.
In the above image you can see one work process removing the protection from a user context, while this same context is locked for all other work processes. Once the work process which removed the protection from the user context performs the roll-out and protects the user context again, this same user context can be now accessed by the other work process.

Parameters and Definitions

\textbf{em/initial\_size\_MB} - This parameter specifies how much SAP extended memory is available on the SAP system.

\textbf{es/max\_seg\_size\_MB} - Specifies the maximum size of an extended memory segment in the SAP system.

\textbf{es/max\_segments} - Specifies the number of extended memory segments in the SAP system.
Memory Implementation "SHM_SEGS"

The parameter ES/TABLE determines the function table used to provide ES functionality. There are two ES/TABLE implementations available. These are UNIX_STD, which is the default, and SHM_SEGS. At present only AIX supports the SHM_SEGS implementation.

Strategy and Characteristics

Parameters and Definitions

**EM/TOTAL_SIZE_MB** - This parameter specifies how much SAP extended memory is available on the SAP system.

**ES/SHM_SEG_SIZE** - Size of the AIX system segments (in megabytes)

**ES/SHM_PROC_SEG_COUNT** - Specifies the number of shared memory segments that are used as shared and privately.

**ES/SHM_MAX_SHARED_SEGS** - Specifies the number of shared memory segments. The value comes from ES/SHM_PROC_SEG_COUNT – ES/SHM_MAX_PRIV_SEGS

**ES/SHM_MAX_PRIV_SEGS** - Specifies the number of shared memory segments that are used privately.

**ES/SHM_USER_COUNT** - The maximum number of users in all open sessions.
ES/SHM_SEG_COUNT - The maximum number of AIX system segments that ESSHM uses.

Related Content

Related Documents

Memory Management on Linux
Implementing the Extended Memory on Windows
Memory Management Under Linux
Implementing the Extended Memory Under UNIX
Old or New Memory Management?
Configuration for AIX
Configuration for AIX
Release of SAP Memory for the Operating System

Related SAP Notes/KBAs

SAP Note 1458065 - Extended memory restricted to 64G
SAP Note 941735 - SAP memory management system for 64-bit Linux systems
SAP Note 1002587 - Flat Memory Model on Windows

(Use the following Hyperlink with SAP Note/KBA number at end: https://service.sap.com/sap/support/notes/123456)