

[PATCH] overcommit symbolic constants

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-09/9045.html>

Andries.Brouwer_at_cwi.nl

Date: 09/30/04

Date: Thu, 30 Sep 2004 15:41:27 +0200 (MEST)
To: akpm@osdl.org, torvalds@osdl.org

Played a bit with overcommit the past hour.
Am not entirely satisfied with the no overcommit mode 2 –
programs segfault when the system is close to that boundary.
So, instead of the somewhat larger patch that I planned to send,
just symbolic names for the modes.

Andries

```
diff -uprN -X /linux/dontdiff a/arch/arm/mm/init.c b/arch/arm/mm/init.c
--- a/arch/arm/mm/init.c 2004-08-26 22:05:07.000000000 +0200
+++ b/arch/arm/mm/init.c 2004-09-30 15:34:51.000000000 +0200
@@ -590,7 +590,7 @@ void __init mem_init(void)
     * anywhere without overcommit, so turn
     * it on by default.
     */
- sysctl_overcommit_memory = 1;
+ sysctl_overcommit_memory = OVERCOMMIT_ALWAYS;
 }
}
```

```
diff -uprN -X /linux/dontdiff a/arch/arm26/mm/init.c b/arch/arm26/mm/init.c
--- a/arch/arm26/mm/init.c 2004-08-26 22:05:07.000000000 +0200
+++ b/arch/arm26/mm/init.c 2004-09-30 15:34:51.000000000 +0200
@@ -376,7 +376,7 @@ void __init mem_init(void)
     * Turn on overcommit on tiny machines
     */
     if (PAGE_SIZE >= 16384 && num_physpages <= 128) {
- sysctl_overcommit_memory = 1;
+ sysctl_overcommit_memory = OVERCOMMIT_ALWAYS;
         printk("Turning on overcommit\n");
     }
 }
```

```
diff -uprN -X /linux/dontdiff a/Documentation/vm/overcommit-accounting
b/Documentation/vm/overcommit-accounting
--- a/Documentation/vm/overcommit-accounting 2003-12-18 03:58:28.000000000 +0100
+++ b/Documentation/vm/overcommit-accounting 2004-09-30 15:34:51.000000000 +0200
@@ -1,4 +1,4 @@
```

Linux-Kernel: [PATCH] overcommit symbolic constants

- The Linux kernel supports three overcommit handling modes
- +The Linux kernel supports the following overcommit handling modes

0 – Heuristic overcommit handling. Obvious overcommits of address space are refused. Used for a typical system. It @@ -7,10 +7,10 @@ The Linux kernel supports three overcomm allocate slightly more memory in this mode. This is the default.

-1 – No overcommit handling. Appropriate for some scientific
+1 – Always overcommit. Appropriate for some scientific applications.

-2 – (NEW) strict overcommit. The total address space commit
+2 – Don't overcommit. The total address space commit for the system is not permitted to exceed swap + a configurable percentage (default is 50) of physical RAM. Depending on the percentage you use, in most situations

@@ -27,7 +27,7 @@ Gotchas

The C language stack growth does an implicit mmap. If you want absolute guarantees and run close to the edge you MUST mmap your stack for the -largest size you think you will need. For typical stack usage is does +largest size you think you will need. For typical stack usage this does not matter much but it's a corner case if you really really care

In mode 2 the MAP_NORESERVE flag is ignored.

```
diff -uprN -X /linux/dontdiff a/include/linux/mman.h b/include/linux/mman.h
```

```
--- a/include/linux/mman.h 2003-12-18 03:58:15.000000000 +0100
```

```
+++ b/include/linux/mman.h 2004-09-30 15:34:51.000000000 +0200
```

```
@@ -10,6 +10,9 @@
```

```
#define MREMAP_MAYMOVE 1
```

```
#define MREMAP_FIXED 2
```

```
+#define OVERCOMMIT_GUESS 0
```

```
+#define OVERCOMMIT_ALWAYS 1
```

```
+#define OVERCOMMIT_NEVER 2
```

```
extern int sysctl_overcommit_memory;
```

```
extern int sysctl_overcommit_ratio;
```

```
extern atomic_t vm_committed_space;
```

```
diff -uprN -X /linux/dontdiff a/mm/mmap.c b/mm/mmap.c
```

```
--- a/mm/mmap.c 2004-08-26 22:05:44.000000000 +0200
```

```
+++ b/mm/mmap.c 2004-09-30 15:35:27.000000000 +0200
```

```
@@ -54,7 +54,7 @@ pgprot_t protection_map[16] = {  
    __S000, __S001, __S010, __S011, __S100, __S101, __S110, __S111  
};
```

```
-int sysctl_overcommit_memory = 0; /* default is heuristic overcommit */
```

```
+int sysctl_overcommit_memory = OVERCOMMIT_GUESS; /* heuristic overcommit */
```

```
int sysctl_overcommit_ratio = 50; /* default is 50% */
```

```
int sysctl_max_map_count = DEFAULT_MAX_MAP_COUNT;
```

[PATCH] overcommit symbolic constants

Linux-Kernel: [PATCH] overcommit symbolic constants

```
atomic_t vm_committed_space = ATOMIC_INIT(0);
@@ -882,7 +882,7 @@ munmap_back:
    return -ENOMEM;

    if (accountable && (!(flags & MAP_NORESERVE) ||
- sysctl_overcommit_memory > 1)) {
+ sysctl_overcommit_memory == OVERCOMMIT_NEVER)) {
        if (vm_flags & VM_SHARED) {
            /* Check memory availability in shmem_file_setup? */
            vm_flags |= VM_ACCOUNT;
diff -uprN -X /linux/dontdiff a/mm/nommu.c b/mm/nommu.c
--- a/mm/nommu.c 2004-08-26 22:05:44.000000000 +0200
+++ b/mm/nommu.c 2004-09-30 15:34:51.000000000 +0200
@@ -30,7 +30,7 @@ unsigned long max_mapnr;
unsigned long num_physpages;
unsigned long askedalloc, realalloc;
atomic_t vm_committed_space = ATOMIC_INIT(0);
-int sysctl_overcommit_memory; /* default is heuristic overcommit */
+int sysctl_overcommit_memory = OVERCOMMIT_GUESS; /* heuristic overcommit */
int sysctl_overcommit_ratio = 50; /* default is 50% */

int sysctl_max_map_count = DEFAULT_MAX_MAP_COUNT;
diff -uprN -X /linux/dontdiff a/security/commoncap.c b/security/commoncap.c
--- a/security/commoncap.c 2004-06-24 17:11:21.000000000 +0200
+++ b/security/commoncap.c 2004-09-30 15:34:51.000000000 +0200
@@ -314,10 +314,10 @@ int cap_vm_enough_memory(long pages)
/*
 * Sometimes we want to use more memory than we have
 */
- if (sysctl_overcommit_memory == 1)
+ if (sysctl_overcommit_memory == OVERCOMMIT_ALWAYS)
    return 0;

- if (sysctl_overcommit_memory == 0) {
+ if (sysctl_overcommit_memory == OVERCOMMIT_GUESS) {
    unsigned long n;

    free = get_page_cache_size();
diff -uprN -X /linux/dontdiff a/security/dummy.c b/security/dummy.c
--- a/security/dummy.c 2004-08-26 22:05:50.000000000 +0200
+++ b/security/dummy.c 2004-09-30 15:34:51.000000000 +0200
@@ -121,10 +121,10 @@ static int dummy_vm_enough_memory(long p
/*
 * Sometimes we want to use more memory than we have
 */
- if (sysctl_overcommit_memory == 1)
+ if (sysctl_overcommit_memory == OVERCOMMIT_ALWAYS)
    return 0;

- if (sysctl_overcommit_memory == 0) {
+ if (sysctl_overcommit_memory == OVERCOMMIT_GUESS) {
```

Linux-Kernel: [PATCH] overcommit symbolic constants

```
    free = get_page_cache_size();
    free += nr_free_pages();
    free += nr_swap_pages;
diff -uprN -X /linux/dontdiff a/security/selinux/hooks.c b/security/selinux/hooks.c
--- a/security/selinux/hooks.c 2004-08-26 22:05:50.000000000 +0200
+++ b/security/selinux/hooks.c 2004-09-30 15:34:51.000000000 +0200
@@ -1548,10 +1548,10 @@ static int selinux_vm_enough_memory(long
/*
 * Sometimes we want to use more memory than we have
 */
- if (sysctl_overcommit_memory == 1)
+ if (sysctl_overcommit_memory == OVERCOMMIT_ALWAYS)
    return 0;

- if (sysctl_overcommit_memory == 0) {
+ if (sysctl_overcommit_memory == OVERCOMMIT_GUESS) {
    free = get_page_cache_size();
    free += nr_free_pages();
    free += nr_swap_pages;
-

```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>