Subject: [patch 1/3] change clone_flags type to u64
Posted by Cedric Le Goater on Thu, 07 Feb 2008 10:31:36 GMT
View Forum Message <> Reply to Message

From: Cedric Le Goater <clg@fr.ibm.com>

This is a preliminary patch changing the clone_flags type to 64bits
for all the routines called by do_fork().

It prepares ground for the next patch which introduces an enhanced
version of clone() supporting 64bits flags.

This is work in progress. All conversions might not be done yet.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
---
 arch/alpha/kernel/process.c | 2 +-
 arch/arm/kernel/process.c | 2 +-
 arch/avr32/kernel/process.c | 2 +-
 arch/blackfin/kernel/process.c | 2 +-
 arch/cris/arch-v10/kernel/process.c | 2 +-
 arch/cris/arch-v32/kernel/process.c | 2 +-
 arch/frv/kernel/process.c | 2 +-
 arch/h8300/kernel/process.c | 2 +-
 arch/ia64/ia32/sys_ia32.c | 2 +-
 arch/ia64/kernel/process.c | 2 +-
 arch/m32r/kernel/process.c | 2 +-
 arch/m68k/kernel/process.c | 2 +-
 arch/m68knommu/kernel/process.c | 2 +-
 arch/mips/kernel/process.c | 2 +-
 arch/mn10300/kernel/process.c | 2 +-
 arch/parisc/kernel/process.c | 2 +-
 arch/powerpc/kernel/process.c | 2 +-
 arch/s390/kernel/process.c | 2 +-
 arch/sh/kernel/process_32.c | 2 +-
 arch/sh/kernel/process_64.c | 2 +-
 arch/sparc/kernel/process.c | 2 +-
 arch/sparc64/kernel/process.c | 2 +-
 arch/um/kernel/process.c | 2 +-
 arch/v850/kernel/process.c | 2 +-
 arch/x86/kernel/process_32.c | 2 +-
 arch/x86/kernel/process_64.c | 2 +-
 arch/xtensa/kernel/process.c | 2 +-
 fs/namespace.c | 2 +-
 include/linux/ipc_namespace.h | 4 ++--
 include/linux/key.h | 2 +-
 include/linux/mnt_namespace.h | 2 +-
 include/linux/nsproxy.h | 4 ++--

```
 include/linux/pid_namespace.h      |    4 ++--
 include/linux/sched.h              |    4 ++--
 include/linux/security.h           |    6 +++---
 include/linux/sem.h                |    4 ++--
 include/linux/user_namespace.h     |    4 ++--
 include/linux/utsname.h            |    4 ++--
 include/net/net_namespace.h        |    4 ++--
 ipc/namespace.c                    |    2 +-
 ipc/sem.c                          |    2 +-
 kernel/fork.c                      |   36 ++++++++++++++++++++++----------------
 kernel/nsproxy.c                   |    6 +++---
 kernel/pid_namespace.c             |    2 +-
 kernel/user_namespace.c            |    2 +-
 kernel/utsname.c                   |    2 +-
 net/core/net_namespace.c           |    4 ++--
 security/dummy.c                   |    2 +-
 security/keys/process_keys.c       |    2 +-
 security/security.c                |    2 +-
 security/selinux/hooks.c           |    2 +-
 51 files changed, 81 insertions(+), 81 deletions(-)

Index: 2.6.24-mm1/arch/alpha/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/alpha/kernel/process.c
+++ 2.6.24-mm1/arch/alpha/kernel/process.c
@@ -270,7 +270,7 @@ alpha_vfork(struct pt_regs *regs)
  */

 int
-copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+copy_thread(int nr, u64 clone_flags, unsigned long usp,
     unsigned long unused,
     struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/arch/arm/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/arm/kernel/process.c
+++ 2.6.24-mm1/arch/arm/kernel/process.c
@@ -331,7 +331,7 @@ void release_thread(struct task_struct *
 asmlinkage void ret_from_fork(void) __asm__("ret_from_fork");

 int
-copy_thread(int nr, unsigned long clone_flags, unsigned long stack_start,
+copy_thread(int nr, u64 clone_flags, unsigned long stack_start,
     unsigned long stk_sz, struct task_struct *p, struct pt_regs *regs)
 {
  struct thread_info *thread = task_thread_info(p);
Index: 2.6.24-mm1/arch/avr32/kernel/process.c
```

```
===================================================================
--- 2.6.24-mm1.orig/arch/avr32/kernel/process.c
+++ 2.6.24-mm1/arch/avr32/kernel/process.c
@@ -325,7 +325,7 @@ int dump_fpu(struct pt_regs *regs, elf_f

 asmlinkage void ret_from_fork(void);

-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
   unsigned long unused,
   struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/blackfin/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/blackfin/kernel/process.c
+++ 2.6.24-mm1/arch/blackfin/kernel/process.c
@@ -168,7 +168,7 @@ asmlinkage int bfin_clone(struct pt_regs
 }

 int
-copy_thread(int nr, unsigned long clone_flags,
+copy_thread(int nr, u64 clone_flags,
      unsigned long usp, unsigned long topstk,
      struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/cris/arch-v10/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/cris/arch-v10/kernel/process.c
+++ 2.6.24-mm1/arch/cris/arch-v10/kernel/process.c
@@ -116,7 +116,7 @@ int kernel_thread(int (*fn)(void *), voi
  */
 asmlinkage void ret_from_fork(void);

-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
   unsigned long unused,
   struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/cris/arch-v32/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/cris/arch-v32/kernel/process.c
+++ 2.6.24-mm1/arch/cris/arch-v32/kernel/process.c
@@ -135,7 +135,7 @@ kernel_thread(int (*fn)(void *), void *
 extern asmlinkage void ret_from_fork(void);

 int
-copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+copy_thread(int nr, u64 clone_flags, unsigned long usp,
```

```
  unsigned long unused,
  struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/frv/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/frv/kernel/process.c
+++ 2.6.24-mm1/arch/frv/kernel/process.c
@@ -204,7 +204,7 @@ void prepare_to_copy(struct task_struct
  /*
   * set up the kernel stack and exception frames for a new process
   */
-int copy_thread(int nr, unsigned long clone_flags,
+int copy_thread(int nr, u64 clone_flags,
    unsigned long usp, unsigned long topstk,
    struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/h8300/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/h8300/kernel/process.c
+++ 2.6.24-mm1/arch/h8300/kernel/process.c
@@ -192,7 +192,7 @@ asmlinkage int h8300_clone(struct pt_reg

 }

-int copy_thread(int nr, unsigned long clone_flags,
+int copy_thread(int nr, u64 clone_flags,
                unsigned long usp, unsigned long topstk,
    struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/arch/ia64/ia32/sys_ia32.c
===================================================================
--- 2.6.24-mm1.orig/arch/ia64/ia32/sys_ia32.c
+++ 2.6.24-mm1/arch/ia64/ia32/sys_ia32.c
@@ -734,7 +734,7 @@ __ia32_copy_pp_list(struct ia64_partial_

 int
 ia32_copy_ia64_partial_page_list(struct task_struct *p,
-    unsigned long clone_flags)
+    u64 clone_flags)
 {
  int retval = 0;

Index: 2.6.24-mm1/arch/ia64/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/ia64/kernel/process.c
+++ 2.6.24-mm1/arch/ia64/kernel/process.c
@@ -401,7 +401,7 @@ ia64_load_extra (struct task_struct *tas
  * so there is nothing to worry about.
```

```
 */
 int
-copy_thread (int nr, unsigned long clone_flags,
+copy_thread (int nr, u64 clone_flags,
       unsigned long user_stack_base, unsigned long user_stack_size,
       struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/m32r/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/m32r/kernel/process.c
+++ 2.6.24-mm1/arch/m32r/kernel/process.c
@@ -242,7 +242,7 @@ int dump_fpu(struct pt_regs *regs, elf_f
  return 0; /* Task didn't use the fpu at all. */
 }

-int copy_thread(int nr, unsigned long clone_flags, unsigned long spu,
+int copy_thread(int nr, u64 clone_flags, unsigned long spu,
  unsigned long unused, struct task_struct *tsk, struct pt_regs *regs)
 {
  struct pt_regs *childregs = task_pt_regs(tsk);
Index: 2.6.24-mm1/arch/m68k/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/m68k/kernel/process.c
+++ 2.6.24-mm1/arch/m68k/kernel/process.c
@@ -235,7 +235,7 @@ asmlinkage int m68k_clone(struct pt_regs
          parent_tidptr, child_tidptr);
 }

-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
    unsigned long unused,
    struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/arch/m68knommu/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/m68knommu/kernel/process.c
+++ 2.6.24-mm1/arch/m68knommu/kernel/process.c
@@ -200,7 +200,7 @@ asmlinkage int m68k_clone(struct pt_regs
         return do_fork(clone_flags, newsp, regs, 0, NULL, NULL);
 }

-int copy_thread(int nr, unsigned long clone_flags,
+int copy_thread(int nr, u64 clone_flags,
   unsigned long usp, unsigned long topstk,
   struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/arch/mips/kernel/process.c
===================================================================
```

```
--- 2.6.24-mm1.orig/arch/mips/kernel/process.c
+++ 2.6.24-mm1/arch/mips/kernel/process.c
@@ -100,7 +100,7 @@ void flush_thread(void)
 {
 }

-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
  unsigned long unused, struct task_struct *p, struct pt_regs *regs)
 {
  struct thread_info *ti = task_thread_info(p);
Index: 2.6.24-mm1/arch/mn10300/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/mn10300/kernel/process.c
+++ 2.6.24-mm1/arch/mn10300/kernel/process.c
@@ -193,7 +193,7 @@ void prepare_to_copy(struct task_struct
  * set up the kernel stack for a new thread and copy arch-specific thread
  * control information
  */
-int copy_thread(int nr, unsigned long clone_flags,
+int copy_thread(int nr, u64 clone_flags,
   unsigned long c_usp, unsigned long ustk_size,
   struct task_struct *p, struct pt_regs *kregs)
 {
Index: 2.6.24-mm1/arch/parisc/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/parisc/kernel/process.c
+++ 2.6.24-mm1/arch/parisc/kernel/process.c
@@ -263,7 +263,7 @@ sys_vfork(struct pt_regs *regs)
 }

 int
-copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+copy_thread(int nr, u64 clone_flags, unsigned long usp,
     unsigned long unused, /* in ia64 this is "user_stack_size" */
     struct task_struct * p, struct pt_regs * pregs)
 {
Index: 2.6.24-mm1/arch/powerpc/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/powerpc/kernel/process.c
+++ 2.6.24-mm1/arch/powerpc/kernel/process.c
@@ -534,7 +534,7 @@ void prepare_to_copy(struct task_struct
 /*
  * Copy a thread..
  */
-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
   unsigned long unused, struct task_struct *p,
```

```
   struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/s390/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/s390/kernel/process.c
+++ 2.6.24-mm1/arch/s390/kernel/process.c
@@ -241,7 +241,7 @@ void release_thread(struct task_struct *
 {
 }

-int copy_thread(int nr, unsigned long clone_flags, unsigned long new_stackp,
+int copy_thread(int nr, u64 clone_flags, unsigned long new_stackp,
   unsigned long unused,
         struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/arch/sh/kernel/process_32.c
===================================================================
--- 2.6.24-mm1.orig/arch/sh/kernel/process_32.c
+++ 2.6.24-mm1/arch/sh/kernel/process_32.c
@@ -232,7 +232,7 @@ int dump_fpu(struct pt_regs *regs, elf_f

 asmlinkage void ret_from_fork(void);

-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
    unsigned long unused,
    struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/sh/kernel/process_64.c
===================================================================
--- 2.6.24-mm1.orig/arch/sh/kernel/process_64.c
+++ 2.6.24-mm1/arch/sh/kernel/process_64.c
@@ -500,7 +500,7 @@ int dump_fpu(struct pt_regs *regs, elf_f

 asmlinkage void ret_from_fork(void);

-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
    unsigned long unused,
    struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/sparc/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/sparc/kernel/process.c
+++ 2.6.24-mm1/arch/sparc/kernel/process.c
@@ -454,7 +454,7 @@ asmlinkage int sparc_do_fork(unsigned lo
  */
 extern void ret_from_fork(void);
```

```
-int copy_thread(int nr, unsigned long clone_flags, unsigned long sp,
+int copy_thread(int nr, u64 clone_flags, unsigned long sp,
   unsigned long unused,
   struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/sparc64/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/sparc64/kernel/process.c
+++ 2.6.24-mm1/arch/sparc64/kernel/process.c
@@ -617,7 +617,7 @@ asmlinkage long sparc_do_fork(unsigned l
  * Parent -->  %o0 == childs  pid, %o1 == 0
  * Child  -->  %o0 == parents pid, %o1 == 1
  */
-int copy_thread(int nr, unsigned long clone_flags, unsigned long sp,
+int copy_thread(int nr, u64 clone_flags, unsigned long sp,
   unsigned long unused,
   struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/um/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/um/kernel/process.c
+++ 2.6.24-mm1/arch/um/kernel/process.c
@@ -181,7 +181,7 @@ void fork_handler(void)
  userspace(&current->thread.regs.regs);
 }

-int copy_thread(int nr, unsigned long clone_flags, unsigned long sp,
+int copy_thread(int nr, u64 clone_flags, unsigned long sp,
   unsigned long stack_top, struct task_struct * p,
   struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/v850/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/v850/kernel/process.c
+++ 2.6.24-mm1/arch/v850/kernel/process.c
@@ -110,7 +110,7 @@ void flush_thread (void)
  set_fs (USER_DS);
 }

-int copy_thread (int nr, unsigned long clone_flags,
+int copy_thread (int nr, u64 clone_flags,
    unsigned long stack_start, unsigned long stack_size,
    struct task_struct *p, struct pt_regs *regs)
 {
Index: 2.6.24-mm1/arch/x86/kernel/process_32.c
===================================================================
--- 2.6.24-mm1.orig/arch/x86/kernel/process_32.c
```

```
+++ 2.6.24-mm1/arch/x86/kernel/process_32.c
@@ -491,7 +491,7 @@ void prepare_to_copy(struct task_struct
  unlazy_fpu(tsk);
 }

-int copy_thread(int nr, unsigned long clone_flags, unsigned long sp,
+int copy_thread(int nr, u64 clone_flags, unsigned long sp,
  unsigned long unused,
  struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/arch/x86/kernel/process_64.c
===================================================================
--- 2.6.24-mm1.orig/arch/x86/kernel/process_64.c
+++ 2.6.24-mm1/arch/x86/kernel/process_64.c
@@ -488,7 +488,7 @@ void prepare_to_copy(struct task_struct
  unlazy_fpu(tsk);
 }

-int copy_thread(int nr, unsigned long clone_flags, unsigned long sp,
+int copy_thread(int nr, u64 clone_flags, unsigned long sp,
   unsigned long unused,
  struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/arch/xtensa/kernel/process.c
===================================================================
--- 2.6.24-mm1.orig/arch/xtensa/kernel/process.c
+++ 2.6.24-mm1/arch/xtensa/kernel/process.c
@@ -102,7 +102,7 @@ void flush_thread(void)
  *       childregs.
  */

-int copy_thread(int nr, unsigned long clone_flags, unsigned long usp,
+int copy_thread(int nr, u64 clone_flags, unsigned long usp,
   unsigned long unused,
                struct task_struct * p, struct pt_regs * regs)
 {
Index: 2.6.24-mm1/include/linux/key.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/key.h
+++ 2.6.24-mm1/include/linux/key.h
@@ -266,7 +266,7 @@ extern struct key root_user_keyring, roo
 extern int alloc_uid_keyring(struct user_struct *user,
        struct task_struct *ctx);
 extern void switch_uid_keyring(struct user_struct *new_user);
-extern int copy_keys(unsigned long clone_flags, struct task_struct *tsk);
+extern int copy_keys(u64 clone_flags, struct task_struct *tsk);
 extern int copy_thread_group_keys(struct task_struct *tsk);
 extern void exit_keys(struct task_struct *tsk);
```

```
 extern void exit_thread_group_keys(struct signal_struct *tg);
Index: 2.6.24-mm1/include/linux/sched.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/sched.h
+++ 2.6.24-mm1/include/linux/sched.h
@@ -1756,7 +1756,7 @@ extern struct mm_struct *get_task_mm(str
 /* Remove the current tasks stale references to the old mm_struct */
 extern void mm_release(struct task_struct *, struct mm_struct *);

-extern int  copy_thread(int, unsigned long, unsigned long, unsigned long, struct task_struct *,
struct pt_regs *);
+extern int  copy_thread(int, u64, unsigned long, unsigned long, struct task_struct *, struct pt_regs
*);
 extern void flush_thread(void);
 extern void exit_thread(void);

@@ -1772,7 +1772,7 @@ extern int allow_signal(int);
 extern int disallow_signal(int);

 extern int do_execve(char *, char __user * __user *, char __user * __user *, struct pt_regs *);
-extern long do_fork(unsigned long, unsigned long, struct pt_regs *, unsigned long, int __user *,
int __user *);
+extern long do_fork(u64, unsigned long, struct pt_regs *, unsigned long, int __user *, int __user
*);
 struct task_struct *fork_idle(int);

 extern void set_task_comm(struct task_struct *tsk, char *from);
Index: 2.6.24-mm1/include/linux/security.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/security.h
+++ 2.6.24-mm1/include/linux/security.h
@@ -1321,7 +1321,7 @@ struct security_operations {
  int (*file_receive) (struct file * file);
  int (*dentry_open)  (struct file *file);

- int (*task_create) (unsigned long clone_flags);
+ int (*task_create) (u64 clone_flags);
  int (*task_alloc_security) (struct task_struct * p);
  void (*task_free_security) (struct task_struct * p);
  int (*task_setuid) (uid_t id0, uid_t id1, uid_t id2, int flags);
@@ -1576,7 +1576,7 @@ int security_file_send_sigiotask(struct
       struct fown_struct *fown, int sig);
 int security_file_receive(struct file *file);
 int security_dentry_open(struct file *file);
-int security_task_create(unsigned long clone_flags);
+int security_task_create(u64 clone_flags);
 int security_task_alloc(struct task_struct *p);
 void security_task_free(struct task_struct *p);
```

```
 int security_task_setuid(uid_t id0, uid_t id1, uid_t id2, int flags);
@@ -2042,7 +2042,7 @@ static inline int security_dentry_open (
  return 0;
 }

-static inline int security_task_create (unsigned long clone_flags)
+static inline int security_task_create (u64 clone_flags)
 {
  return 0;
 }
Index: 2.6.24-mm1/include/linux/sem.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/sem.h
+++ 2.6.24-mm1/include/linux/sem.h
@@ -139,11 +139,11 @@ struct sysv_sem {

 #ifdef CONFIG_SYSVIPC

-extern int copy_semundo(unsigned long clone_flags, struct task_struct *tsk);
+extern int copy_semundo(u64 clone_flags, struct task_struct *tsk);
 extern void exit_sem(struct task_struct *tsk);

 #else
-static inline int copy_semundo(unsigned long clone_flags, struct task_struct *tsk)
+static inline int copy_semundo(u64 clone_flags, struct task_struct *tsk)
 {
  return 0;
 }
Index: 2.6.24-mm1/ipc/sem.c
===================================================================
--- 2.6.24-mm1.orig/ipc/sem.c
+++ 2.6.24-mm1/ipc/sem.c
@@ -1260,7 +1260,7 @@ asmlinkage long sys_semop (int semid, st
  * parent and child tasks.
  */

-int copy_semundo(unsigned long clone_flags, struct task_struct *tsk)
+int copy_semundo(u64 clone_flags, struct task_struct *tsk)
 {
  struct sem_undo_list *undo_list;
  int error;
Index: 2.6.24-mm1/kernel/fork.c
===================================================================
--- 2.6.24-mm1.orig/kernel/fork.c
+++ 2.6.24-mm1/kernel/fork.c
@@ -547,7 +547,7 @@ fail_nocontext:
  return NULL;
 }
```

```
-static int copy_mm(unsigned long clone_flags, struct task_struct * tsk)
+static int copy_mm(u64 clone_flags, struct task_struct * tsk)
 {
  struct mm_struct * mm, *oldmm;
  int retval;
@@ -623,7 +623,7 @@ struct fs_struct *copy_fs_struct(struct

 EXPORT_SYMBOL_GPL(copy_fs_struct);

-static int copy_fs(unsigned long clone_flags, struct task_struct *tsk)
+static int copy_fs(u64 clone_flags, struct task_struct *tsk)
 {
  if (clone_flags & CLONE_FS) {
   atomic_inc(&current->fs->count);
@@ -765,7 +765,7 @@ out:
  return NULL;
 }

-static int copy_files(unsigned long clone_flags, struct task_struct * tsk)
+static int copy_files(u64 clone_flags, struct task_struct * tsk)
 {
  struct files_struct *oldf, *newf;
  int error = 0;
@@ -798,7 +798,7 @@ out:
  return error;
 }

-static int copy_io(unsigned long clone_flags, struct task_struct *tsk)
+static int copy_io(u64 clone_flags, struct task_struct *tsk)
 {
 #ifdef CONFIG_BLOCK
  struct io_context *ioc = current->io_context;
@@ -851,7 +851,7 @@ int unshare_files(void)

 EXPORT_SYMBOL(unshare_files);

-static int copy_sighand(unsigned long clone_flags, struct task_struct *tsk)
+static int copy_sighand(u64 clone_flags, struct task_struct *tsk)
 {
  struct sighand_struct *sig;

@@ -874,7 +874,7 @@ void __cleanup_sighand(struct sighand_st
   kmem_cache_free(sighand_cachep, sighand);
 }

-static int copy_signal(unsigned long clone_flags, struct task_struct *tsk)
+static int copy_signal(u64 clone_flags, struct task_struct *tsk)
```

```
 {
  struct signal_struct *sig;
  int ret;
@@ -965,7 +965,7 @@ static void cleanup_signal(struct task_s
    __cleanup_signal(sig);
 }

-static void copy_flags(unsigned long clone_flags, struct task_struct *p)
+static void copy_flags(u64 clone_flags, struct task_struct *p)
 {
  unsigned long new_flags = p->flags;

@@ -1001,7 +1001,7 @@ static void rt_mutex_init_task(struct ta
  * parts of the process environment (as per the clone
  * flags). The actual kick-off is left to the caller.
  */
-static struct task_struct *copy_process(unsigned long clone_flags,
+static struct task_struct *copy_process(u64 clone_flags,
       unsigned long stack_start,
       struct pt_regs *regs,
       unsigned long stack_size,
@@ -1423,7 +1423,7 @@ struct task_struct * __cpuinit fork_idle
  return task;
 }

-static int fork_traceflag(unsigned clone_flags)
+static int fork_traceflag(u64 clone_flags)
 {
  if (clone_flags & CLONE_UNTRACED)
   return 0;
@@ -1445,7 +1445,7 @@ static int fork_traceflag(unsigned clone
  * It copies the process, and if successful kick-starts
  * it and waits for it to finish using the VM if required.
  */
-long do_fork(unsigned long clone_flags,
+long do_fork(u64 clone_flags,
         unsigned long stack_start,
         struct pt_regs *regs,
         unsigned long stack_size,
@@ -1467,7 +1467,7 @@ long do_fork(unsigned long clone_flags,

    count--;
    printk(KERN_INFO "fork(): process `%s' used deprecated "
-     "clone flags 0x%lx\n",
+     "clone flags 0x%llx\n",
      get_task_comm(comm, current),
      clone_flags & CLONE_STOPPED);
   }
```

```
@@ -1570,7 +1570,7 @@ void __init proc_caches_init(void)
  * Check constraints on flags passed to the unshare system call and
  * force unsharing of additional process context as appropriate.
  */
-static void check_unshare_flags(unsigned long *flags_ptr)
+static void check_unshare_flags(u64 *flags_ptr)
 {
  /*
   * If unsharing a thread from a thread group, must also
@@ -1603,7 +1603,7 @@ static void check_unshare_flags(unsigned
 /*
  * Unsharing of tasks created with CLONE_THREAD is not supported yet
  */
-static int unshare_thread(unsigned long unshare_flags)
+static int unshare_thread(u64 unshare_flags)
 {
  if (unshare_flags & CLONE_THREAD)
   return -EINVAL;
@@ -1614,7 +1614,7 @@ static int unshare_thread(unsigned long
 /*
  * Unshare the filesystem structure if it is being shared
  */
-static int unshare_fs(unsigned long unshare_flags, struct fs_struct **new_fsp)
+static int unshare_fs(u64 unshare_flags, struct fs_struct **new_fsp)
 {
  struct fs_struct *fs = current->fs;

@@ -1631,7 +1631,7 @@ static int unshare_fs(unsigned long unsh
 /*
  * Unsharing of sighand is not supported yet
  */
-static int unshare_sighand(unsigned long unshare_flags, struct sighand_struct **new_sighp)
+static int unshare_sighand(u64 unshare_flags, struct sighand_struct **new_sighp)
 {
  struct sighand_struct *sigh = current->sighand;

@@ -1644,7 +1644,7 @@ static int unshare_sighand(unsigned long
 /*
  * Unshare vm if it is being shared
  */
-static int unshare_vm(unsigned long unshare_flags, struct mm_struct **new_mmp)
+static int unshare_vm(u64 unshare_flags, struct mm_struct **new_mmp)
 {
  struct mm_struct *mm = current->mm;

@@ -1659,7 +1659,7 @@ static int unshare_vm(unsigned long unsh
 /*
  * Unshare file descriptor table if it is being shared
```

```
 */
-static int unshare_fd(unsigned long unshare_flags, struct files_struct **new_fdp)
+static int unshare_fd(u64 unshare_flags, struct files_struct **new_fdp)
 {
  struct files_struct *fd = current->files;
  int error = 0;
@@ -1678,7 +1678,7 @@ static int unshare_fd(unsigned long unsh
  * Unsharing of semundo for tasks created with CLONE_SYSVSEM is not
  * supported yet
  */
-static int unshare_semundo(unsigned long unshare_flags, struct sem_undo_list **new_ulistp)
+static int unshare_semundo(u64 unshare_flags, struct sem_undo_list **new_ulistp)
 {
  if (unshare_flags & CLONE_SYSVSEM)
   return -EINVAL;
Index: 2.6.24-mm1/security/dummy.c
===================================================================
--- 2.6.24-mm1.orig/security/dummy.c
+++ 2.6.24-mm1/security/dummy.c
@@ -493,7 +493,7 @@ static int dummy_dentry_open (struct fil
  return 0;
 }

-static int dummy_task_create (unsigned long clone_flags)
+static int dummy_task_create (u64 clone_flags)
 {
  return 0;
 }
Index: 2.6.24-mm1/security/keys/process_keys.c
===================================================================
--- 2.6.24-mm1.orig/security/keys/process_keys.c
+++ 2.6.24-mm1/security/keys/process_keys.c
@@ -278,7 +278,7 @@ int copy_thread_group_keys(struct task_s
 /*
  * copy the keys for fork
  */
-int copy_keys(unsigned long clone_flags, struct task_struct *tsk)
+int copy_keys(u64 clone_flags, struct task_struct *tsk)
 {
  key_check(tsk->thread_keyring);
  key_check(tsk->request_key_auth);
Index: 2.6.24-mm1/security/security.c
===================================================================
--- 2.6.24-mm1.orig/security/security.c
+++ 2.6.24-mm1/security/security.c
@@ -578,7 +578,7 @@ int security_dentry_open(struct file *fi
  return security_ops->dentry_open(file);
 }
```

```
-int security_task_create(unsigned long clone_flags)
+int security_task_create(u64 clone_flags)
 {
  return security_ops->task_create(clone_flags);
 }
Index: 2.6.24-mm1/security/selinux/hooks.c
===================================================================
--- 2.6.24-mm1.orig/security/selinux/hooks.c
+++ 2.6.24-mm1/security/selinux/hooks.c
@@ -3019,7 +3019,7 @@ static int selinux_dentry_open(struct fi

 /* task security operations */

-static int selinux_task_create(unsigned long clone_flags)
+static int selinux_task_create(u64 clone_flags)
 {
  int rc;

Index: 2.6.24-mm1/include/linux/nsproxy.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/nsproxy.h
+++ 2.6.24-mm1/include/linux/nsproxy.h
@@ -62,11 +62,11 @@ static inline struct nsproxy *task_nspro
  return rcu_dereference(tsk->nsproxy);
 }

-int copy_namespaces(unsigned long flags, struct task_struct *tsk);
+int copy_namespaces(u64 clone_flags, struct task_struct *tsk);
 void exit_task_namespaces(struct task_struct *tsk);
 void switch_task_namespaces(struct task_struct *tsk, struct nsproxy *new);
 void free_nsproxy(struct nsproxy *ns);
-int unshare_nsproxy_namespaces(unsigned long, struct nsproxy **,
+int unshare_nsproxy_namespaces(u64, struct nsproxy **,
  struct fs_struct *);

 static inline void put_nsproxy(struct nsproxy *ns)
Index: 2.6.24-mm1/kernel/nsproxy.c
===================================================================
--- 2.6.24-mm1.orig/kernel/nsproxy.c
+++ 2.6.24-mm1/kernel/nsproxy.c
@@ -47,7 +47,7 @@ static inline struct nsproxy *clone_nspr
  * Return the newly created nsproxy.  Do not attach this to the task,
  * leave it to the caller to do proper locking and attach it to task.
  */
-static struct nsproxy *create_new_namespaces(unsigned long flags,
+static struct nsproxy *create_new_namespaces(u64 flags,
    struct task_struct *tsk, struct fs_struct *new_fs)
```

```
 {
  struct nsproxy *new_nsp;
@@ -119,7 +119,7 @@ out_ns:
  * called from clone.  This now handles copy for nsproxy and all
  * namespaces therein.
  */
-int copy_namespaces(unsigned long flags, struct task_struct *tsk)
+int copy_namespaces(u64 flags, struct task_struct *tsk)
 {
  struct nsproxy *old_ns = tsk->nsproxy;
  struct nsproxy *new_ns;
@@ -178,7 +178,7 @@ void free_nsproxy(struct nsproxy *ns)
  * Called from unshare. Unshare all the namespaces part of nsproxy.
  * On success, returns the new nsproxy.
  */
-int unshare_nsproxy_namespaces(unsigned long unshare_flags,
+int unshare_nsproxy_namespaces(u64 unshare_flags,
   struct nsproxy **new_nsp, struct fs_struct *new_fs)
 {
  int err = 0;
Index: 2.6.24-mm1/fs/namespace.c
===================================================================
--- 2.6.24-mm1.orig/fs/namespace.c
+++ 2.6.24-mm1/fs/namespace.c
@@ -1958,7 +1958,7 @@ static struct mnt_namespace *dup_mnt_ns(
  return new_ns;
 }

-struct mnt_namespace *copy_mnt_ns(unsigned long flags, struct mnt_namespace *ns,
+struct mnt_namespace *copy_mnt_ns(u64 flags, struct mnt_namespace *ns,
   struct fs_struct *new_fs)
 {
  struct mnt_namespace *new_ns;
Index: 2.6.24-mm1/include/linux/ipc_namespace.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/ipc_namespace.h
+++ 2.6.24-mm1/include/linux/ipc_namespace.h
@@ -42,7 +42,7 @@ extern struct ipc_namespace init_ipc_ns;

 #if defined(CONFIG_SYSVIPC) && defined(CONFIG_IPC_NS)
 extern void free_ipc_ns(struct kref *kref);
-extern struct ipc_namespace *copy_ipcs(unsigned long flags,
+extern struct ipc_namespace *copy_ipcs(u64 flags,
           struct ipc_namespace *ns);
 extern void free_ipcs(struct ipc_namespace *ns, struct ipc_ids *ids,
        void (*free)(struct ipc_namespace *,
@@ -60,7 +60,7 @@ static inline void put_ipc_ns(struct ipc
  kref_put(&ns->kref, free_ipc_ns);
```

```
 }
 #else
-static inline struct ipc_namespace *copy_ipcs(unsigned long flags,
+static inline struct ipc_namespace *copy_ipcs(u64 flags,
    struct ipc_namespace *ns)
 {
  if (flags & CLONE_NEWIPC)
Index: 2.6.24-mm1/include/linux/mnt_namespace.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/mnt_namespace.h
+++ 2.6.24-mm1/include/linux/mnt_namespace.h
@@ -14,7 +14,7 @@ struct mnt_namespace {
  int event;
 };

-extern struct mnt_namespace *copy_mnt_ns(unsigned long, struct mnt_namespace *,
+extern struct mnt_namespace *copy_mnt_ns(u64, struct mnt_namespace *,
    struct fs_struct *);
 extern void __put_mnt_ns(struct mnt_namespace *ns);

Index: 2.6.24-mm1/include/linux/pid_namespace.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/pid_namespace.h
+++ 2.6.24-mm1/include/linux/pid_namespace.h
@@ -37,7 +37,7 @@ static inline struct pid_namespace *get_
  return ns;
 }

-extern struct pid_namespace *copy_pid_ns(unsigned long flags, struct pid_namespace *ns);
+extern struct pid_namespace *copy_pid_ns(u64 flags, struct pid_namespace *ns);
 extern void free_pid_ns(struct kref *kref);
 extern void zap_pid_ns_processes(struct pid_namespace *pid_ns);

@@ -56,7 +56,7 @@ static inline struct pid_namespace *get_
 }

 static inline struct pid_namespace *
-copy_pid_ns(unsigned long flags, struct pid_namespace *ns)
+copy_pid_ns(u64 flags, struct pid_namespace *ns)
 {
  if (flags & CLONE_NEWPID)
    ns = ERR_PTR(-EINVAL);
Index: 2.6.24-mm1/include/linux/user_namespace.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/user_namespace.h
+++ 2.6.24-mm1/include/linux/user_namespace.h
@@ -26,7 +26,7 @@ static inline struct user_namespace *get
  return ns;
```

```
 }

-extern struct user_namespace *copy_user_ns(int flags,
+extern struct user_namespace *copy_user_ns(u64 flags,
         struct user_namespace *old_ns);
 extern void free_user_ns(struct kref *kref);

@@ -43,7 +43,7 @@ static inline struct user_namespace *get
  return &init_user_ns;
 }

-static inline struct user_namespace *copy_user_ns(int flags,
+static inline struct user_namespace *copy_user_ns(u64 flags,
         struct user_namespace *old_ns)
 {
  if (flags & CLONE_NEWUSER)
Index: 2.6.24-mm1/include/linux/utsname.h
===================================================================
--- 2.6.24-mm1.orig/include/linux/utsname.h
+++ 2.6.24-mm1/include/linux/utsname.h
@@ -50,7 +50,7 @@ static inline void get_uts_ns(struct uts
  kref_get(&ns->kref);
 }

-extern struct uts_namespace *copy_utsname(unsigned long flags,
+extern struct uts_namespace *copy_utsname(u64 flags,
       struct uts_namespace *ns);
 extern void free_uts_ns(struct kref *kref);

@@ -67,7 +67,7 @@ static inline void put_uts_ns(struct uts
 {
 }

-static inline struct uts_namespace *copy_utsname(unsigned long flags,
+static inline struct uts_namespace *copy_utsname(u64 flags,
       struct uts_namespace *ns)
 {
  if (flags & CLONE_NEWUTS)
Index: 2.6.24-mm1/include/net/net_namespace.h
===================================================================
--- 2.6.24-mm1.orig/include/net/net_namespace.h
+++ 2.6.24-mm1/include/net/net_namespace.h
@@ -73,9 +73,9 @@ extern struct net init_net;
 extern struct list_head net_namespace_list;

 #ifdef CONFIG_NET
-extern struct net *copy_net_ns(unsigned long flags, struct net *net_ns);
+extern struct net *copy_net_ns(u64 flags, struct net *net_ns);
```

```
 #else
-static inline struct net *copy_net_ns(unsigned long flags, struct net *net_ns)
+static inline struct net *copy_net_ns(u64 flags, struct net *net_ns)
 {
  /* There is nothing to copy so this is a noop */
  return net_ns;
Index: 2.6.24-mm1/ipc/namespace.c
===================================================================
--- 2.6.24-mm1.orig/ipc/namespace.c
+++ 2.6.24-mm1/ipc/namespace.c
@@ -28,7 +28,7 @@ static struct ipc_namespace *clone_ipc_n
  return ns;
 }

-struct ipc_namespace *copy_ipcs(unsigned long flags, struct ipc_namespace *ns)
+struct ipc_namespace *copy_ipcs(u64 flags, struct ipc_namespace *ns)
 {
  struct ipc_namespace *new_ns;

Index: 2.6.24-mm1/kernel/pid_namespace.c
===================================================================
--- 2.6.24-mm1.orig/kernel/pid_namespace.c
+++ 2.6.24-mm1/kernel/pid_namespace.c
@@ -115,7 +115,7 @@ static void destroy_pid_namespace(struct
  kmem_cache_free(pid_ns_cachep, ns);
 }

-struct pid_namespace *copy_pid_ns(unsigned long flags, struct pid_namespace *old_ns)
+struct pid_namespace *copy_pid_ns(u64 flags, struct pid_namespace *old_ns)
 {
  struct pid_namespace *new_ns;

Index: 2.6.24-mm1/kernel/user_namespace.c
===================================================================
--- 2.6.24-mm1.orig/kernel/user_namespace.c
+++ 2.6.24-mm1/kernel/user_namespace.c
@@ -49,7 +49,7 @@ static struct user_namespace *clone_user
  return ns;
 }

-struct user_namespace * copy_user_ns(int flags, struct user_namespace *old_ns)
+struct user_namespace * copy_user_ns(u64 flags, struct user_namespace *old_ns)
 {
  struct user_namespace *new_ns;

Index: 2.6.24-mm1/kernel/utsname.c
===================================================================
--- 2.6.24-mm1.orig/kernel/utsname.c
```

```
+++ 2.6.24-mm1/kernel/utsname.c
@@ -41,7 +41,7 @@ static struct uts_namespace *clone_uts_n
  * utsname of this process won't be seen by parent, and vice
  * versa.
  */
-struct uts_namespace *copy_utsname(unsigned long flags, struct uts_namespace *old_ns)
+struct uts_namespace *copy_utsname(u64 flags, struct uts_namespace *old_ns)
 {
  struct uts_namespace *new_ns;

Index: 2.6.24-mm1/net/core/net_namespace.c
===================================================================
--- 2.6.24-mm1.orig/net/core/net_namespace.c
+++ 2.6.24-mm1/net/core/net_namespace.c
@@ -79,7 +79,7 @@ static void net_free(struct net *net)
  kmem_cache_free(net_cachep, net);
 }

-struct net *copy_net_ns(unsigned long flags, struct net *old_net)
+struct net *copy_net_ns(u64 flags, struct net *old_net)
 {
  struct net *new_net = NULL;
  int err;
@@ -155,7 +155,7 @@ void __put_net(struct net *net)
 EXPORT_SYMBOL_GPL(__put_net);

 #else
-struct net *copy_net_ns(unsigned long flags, struct net *old_net)
+struct net *copy_net_ns(u64 flags, struct net *old_net)
 {
  if (flags & CLONE_NEWNET)
   return ERR_PTR(-EINVAL);

--

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers
```