
Subject: Re: [PATCH] Add kernel/notifier.c

Posted by [Andrew Morton](#) on Thu, 19 Jul 2007 21:48:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thu, 19 Jul 2007 20:46:11 +0400

Alexey Dobriyan <adobriyan@sw.ru> wrote:

> There is separate notifier header, but no separate notifier .c file.

>

> Extract notifier code out of kernel/sys.c which will remain for

> misc syscalls I hope. Merge kernel/die_notifier.c into kernel/notifier.c.

If you were running checkpatch (I hope you are) then you'd find that we copied over a whole pile of cruft. We might as well fix that up while we're moving the code around.

Also, your patch tried to add some trailing whitespace, but checkpatch failed to notice that.

(The code motion and the cleanups should really be separate patches, and indeed they are, but I'll end up joining them before it hits git)

```
diff -puN kernel/notifier.c~add-kernel-notifierc-fix kernel/notifier.c
```

```
--- a/kernel/notifier.c~add-kernel-notifierc-fix
```

```
+++ a/kernel/notifier.c
```

```
@@ -107,7 +107,6 @@ int atomic_notifier_chain_register(struct  
    spin_unlock_irqrestore(&nh->lock, flags);  
    return ret;  
}
```

```
-
```

```
EXPORT_SYMBOL_GPL(atomic_notifier_chain_register);
```

```
/**
```

```
@@ -131,7 +130,6 @@ int atomic_notifier_chain_unregister(struct  
    synchronize_rcu();  
    return ret;  
}
```

```
-
```

```
EXPORT_SYMBOL_GPL(atomic_notifier_chain_unregister);
```

```
/**
```

```
@@ -165,7 +163,6 @@ int __kprobes __atomic_notifier_call_chain  
    rcu_read_unlock();  
    return ret;  
}
```

```
-
```

```
EXPORT_SYMBOL_GPL(__atomic_notifier_call_chain);
```

```

int __kprobes atomic_notifier_call_chain(struct atomic_notifier_head *nh,
@@ -173,8 +170,8 @@ int __kprobes atomic_notifier_call_chain
{
    return __atomic_notifier_call_chain(nh, val, v, -1, NULL);
}
-
EXPORT_SYMBOL_GPL(atomic_notifier_call_chain);
+
/*
 * Blocking notifier chain routines. All access to the chain is
 * synchronized by an rwsem.
@@ -209,7 +206,6 @@ int blocking_notifier_chain_register(str
    up_write(&nh->rwsem);
    return ret;
}
-
EXPORT_SYMBOL_GPL(blocking_notifier_chain_register);

/**
@@ -240,7 +236,6 @@ int blocking_notifier_chain_unregister(s
    up_write(&nh->rwsem);
    return ret;
}
-
EXPORT_SYMBOL_GPL(blocking_notifier_chain_unregister);

/**
@@ -311,7 +306,6 @@ int raw_notifier_chain_register(struct r
{
    return notifier_chain_register(&nh->head, n);
}
-
EXPORT_SYMBOL_GPL(raw_notifier_chain_register);

/**
@@ -329,7 +323,6 @@ int raw_notifier_chain_unregister(struct
{
    return notifier_chain_unregister(&nh->head, n);
}
-
EXPORT_SYMBOL_GPL(raw_notifier_chain_unregister);

/**
@@ -358,7 +351,6 @@ int __raw_notifier_call_chain(struct raw
{
    return notifier_call_chain(&nh->head, val, v, nr_to_call, nr_calls);
}
-

```

```

EXPORT_SYMBOL_GPL(__raw_notifier_call_chain);

int raw_notifier_call_chain(struct raw_notifier_head *nh,
@@ -366,7 +358,6 @@ int raw_notifier_call_chain(struct raw_n
{
    return __raw_notifier_call_chain(nh, val, v, -1, NULL);
}
-
EXPORT_SYMBOL_GPL(raw_notifier_call_chain);

/*
@@ -403,7 +394,6 @@ int srcu_notifier_chain_register(struct
    mutex_unlock(&nh->mutex);
    return ret;
}
-
EXPORT_SYMBOL_GPL(srcu_notifier_chain_register);

/**
@@ -435,7 +425,6 @@ int srcu_notifier_chain_unregister(struc
    synchronize_srcu(&nh->srcu);
    return ret;
}
-
EXPORT_SYMBOL_GPL(srcu_notifier_chain_unregister);

/**
@@ -498,7 +487,6 @@ void srcu_init_notifier_head(struct srcu
    BUG();
    nh->head = NULL;
}
-
EXPORT_SYMBOL_GPL(srcu_init_notifier_head);

/**
@@ -512,11 +500,10 @@ EXPORT_SYMBOL_GPL(srcu_init_notifier_he
 * always returns zero.
 */

-int register_reboot_notifier(struct notifier_block * nb)
+int register_reboot_notifier(struct notifier_block *nb)
{
    return blocking_notifier_chain_register(&reboot_notifier_list, nb);
}
-
EXPORT_SYMBOL(register_reboot_notifier);

/**

```

```

@@ -529,15 +516,13 @@ EXPORT_SYMBOL(register_reboot_notifier);
 * Returns zero on success, or %-ENOENT on failure.
 */

-int unregister_reboot_notifier(struct notifier_block * nb)
+int unregister_reboot_notifier(struct notifier_block *nb)
{
    return blocking_notifier_chain_unregister(&reboot_notifier_list , nb);
}
-
EXPORT_SYMBOL(unregister_reboot_notifier);

-
static ATOMIC_NOTIFIER_HEAD(die_chain);

int notify_die(enum die_val val, const char *str,

```
