

~¥½Ä¼ÄfÄ,Ä%ÄfÄ Ä®Ä!Ä%Ä<Ä!Ä³Ä•ÄfÄ Ä®Ä~ÄfÄCEÄ!Ä™Ä~ÄfÄ Ä«ÄfÄ Ä,ÄfÄ,Ä<Ä¥Ä...Ä±Ä!ÄœÄ%
 ÄfÄ,ÄCEÄfÄ Ä~ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«Ä©Ä-Ä<Ä§Ä™Ä°ÄfÄ Ä®Ä¥Ä^Ä ¥Ä¿ÄfÄ~Ä€Ä...ÄfÄ ÄªÄ
 -
 Ä©Ä‡Ä Ä~Ä!Ä ÄfÄ Ä§ÄfÄ Ä™ÄfÄ€Ä,ÄfÄ ÄªÄfÄ ÄœÄfÄ ÄªÄfÄ,Ä%ÄfÄ Ä“ÄfÄ Ä®ÄfÄfÄ%ÄfÄ,Ä-ÄfÄ
 +
 Ä©Ä‡Ä Ä~Ä!Ä ÄfÄ Ä§ÄfÄ Ä™ÄfÄ€Ä,ÄfÄ ÄªÄfÄ ÄœÄfÄ ÄªÄfÄ,Ä%ÄfÄ Ä“ÄfÄ Ä®ÄfÄfÄ%ÄfÄ,Ä-ÄfÄ
 Ä~Ä¿ÄCEÄ¥Ä<Ä•ÄfÄ Ä«ÄfÄ ÄªÄfÄ Ä,,ÄfÄ Ä!ÄfÄ Ä®Ä¥ÄªÄšÄfÄ Ä ÄfÄ Ä®Ä~ÄªÄ~Ä§ÄfÄÄÄ,Ä,,Ä!Ä.Ä

Documentation/stable_kernel_rules.txt

ÄfÄ Ä“ÄfÄ Ä®ÄfÄfÄ•ÄfÄ,Ä¿ÄfÄ,ÄªÄfÄfÄ«ÄfÄ Ä~ÄfÄ Ä©ÄfÄ Ä®ÄfÄ,Ä~ÄfÄ Ä‡ÄfÄ Ä<
 stable
 ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«ÄfÄ Ä®ÄfÄfÄªÄfÄªÄfÄ¼ÄfÄ,Ä¹ÄfÄ ÄCEÄ~Ä¿ÄCEÄfÄ,Ä ÄfÄ,ÄCEÄ
 ÄfÄfÄ«ÄfÄ ÄCEÄ~Ä~Ä~Ä¿Ä°ÄfÄ Ä•ÄfÄ,ÄCEÄfÄ Ä!ÄfÄ Ä,,ÄfÄ Ä¾ÄfÄ Ä™ÄfÄ€Ä,ÄfÄ Ä ÄfÄ Ä—ÄfÄ
 -
 ÄfÄ,ÄšÄ¥Ä...Ä¥ÄfÄ,ÄCEÄfÄ Ä!ÄfÄ,Ä,ÄfÄ,Ä%ÄfÄ Ä,ÄfÄ ÄÿÄfÄ Ä,,Ä¥Ä Ä~Ä¥Ä Ä~ÄfÄ Ä«ÄªÄ½Ä•ÄfÄ
 +
 ÄfÄ,ÄšÄ¥Ä...Ä¥ÄfÄ,ÄCEÄfÄ Ä!ÄfÄ,Ä,ÄfÄ,Ä%ÄfÄ Ä,ÄfÄ ÄÿÄfÄ Ä,,Ä¥Ä Ä~Ä¥Ä Ä~ÄfÄ Ä«ÄªÄ½Ä•ÄfÄ

Documentation/kernel-docs.txt

ÄfÄ€Ä€ÄfÄ€Ä€ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«Ä©Ä-Ä<Ä§Ä™Ä°ÄfÄ Ä«ÄªÄ»Ä~Ä©ÄšÄ ÄfÄ Ä™ÄfÄ,Ä
 @@ -218,9 +220,9 @@ web
 ÄfÄ,ÄªÄfÄ,ÄªÄfÄfÄ^ÄfÄ Ä«ÄfÄ Ä~ÄfÄ€Ä ÄfÄ,Ä³ÄfÄfÄ¼ÄfÄfÄ%ÄfÄ Ä®Ä!Ä§Ä<Ä!Ä^Ä ÄfÄ€Ä ÄfÄ
 ÄfÄ Ä“ÄfÄ Ä“ÄfÄ Ä«ÄfÄ Ä~ÄfÄ€Ä ÄfÄ Ä¾ÄfÄ ÄÿÄfÄ€Ä ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«ÄfÄ Ä®ÄfÄ
 Ä§ÄšÄ,,ÄfÄ ÄªÄ¥ÄÿÄ°Ä!ÄœÄ-Ä!ÄfÄ...Ä¥Ä Ä±ÄfÄ,Ä,Ä~Ä~Ä~Ä¿Ä°ÄfÄ Ä•ÄfÄ,ÄCEÄfÄ Ä!ÄfÄ Ä,,ÄfÄ Ä
 -ÄfÄ Ä,ÄfÄ ÄªÄfÄ ÄÿÄfÄ ÄCEÄfÄ Ä©ÄfÄ Ä“ÄfÄ Ä<ÄfÄ,Ä%ÄfÄ,Ä¹ÄfÄ,Ä¿ÄfÄfÄ¼ÄfÄfÄ^ÄfÄ Ä—Ä
 ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«Ä©Ä-Ä<Ä§Ä™Ä°ÄfÄ,Ä³ÄfÄfÄÿÄfÄfÄ¥
 +ÄfÄ Ä,ÄfÄ ÄªÄfÄ ÄÿÄfÄ ÄCEÄfÄ Ä©ÄfÄ Ä“ÄfÄ Ä<ÄfÄ,Ä%ÄfÄ,Ä¹ÄfÄ,Ä¿ÄfÄfÄ¼ÄfÄfÄ^ÄfÄ Ä—Ä
 ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«Ä©Ä-Ä<Ä§Ä™Ä°ÄfÄ,Ä³ÄfÄfÄÿÄfÄfÄ¥
 ÄfÄfÄ«ÄfÄfÄ‡ÄfÄ,ÄfÄfÄ Ä«Ä¥Ä Ä,Ä¥ÄšÄ ÄfÄ Ä—ÄfÄ Ä!ÄªÄ½Ä•ÄfÄ Ä<ÄfÄ Ä™ÄfÄ,Ä<ÄfÄ Ä“ÄfÄ

kernel

-Janitor's

ÄfÄfÄ—ÄfÄfÄ~ÄfÄ,Ä,ÄfÄ,Ä§ÄfÄ,Ä~ÄfÄfÄ^ÄfÄ Ä«ÄfÄ Ä,,ÄfÄ Ä~ÄfÄ Ä°ÄfÄ,Ä~ÄfÄ Ä,,ÄfÄ Ä§ÄfÄ Ä
 -
 +Janitor's
 ÄfÄfÄ—ÄfÄfÄ~ÄfÄ,Ä,ÄfÄ,Ä§ÄfÄ,Ä~ÄfÄfÄ^ÄfÄ Ä«ÄfÄ Ä,ÄfÄ Ä~ÄfÄ Ä°Ä~Ä%Ä~ÄfÄ Ä,,ÄfÄ Ä§ÄfÄ

<http://janitor.kernelnewbies.org/>

ÄfÄ Ä“ÄfÄ Ä“ÄfÄ Ä~ÄfÄ Ä ÄfÄ Ä®ÄfÄ,Ä~ÄfÄ Ä‡ÄfÄ ÄªÄfÄ,Ä¹ÄfÄ,Ä¿ÄfÄfÄ¼ÄfÄfÄ^ÄfÄ,Ä~ÄfÄ Ä™ÄfÄ
 Linux

ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«ÄfÄ,Ä½ÄfÄfÄ¼ÄfÄ,Ä¹ÄfÄfÄ,,ÄfÄfÄªÄfÄ¼ÄfÄ Ä®ÄªÄ,Ä-ÄfÄ Ä«Ä
 @@ -243,7 +245,7 @@ Linux

ÄfÄ,Ä«ÄfÄfÄ¼ÄfÄfÄ ÄfÄfÄ«ÄfÄ,Ä½ÄfÄfÄ¼ÄfÄ,Ä¹ÄfÄfÄ,,ÄfÄfÄªÄfÄ¼ÄfÄ Ä®ÄªÄ,Ä-ÄfÄ Ä«Ä
 Ä~Ä‡ÄªÄ¥Ä.Ä±Ä¥Ä Ä,Ä§Ä...Ä§Ä!Ä-Ä¹Ä¥Ä¼Ä ÄfÄ Ä§ÄfÄ€Ä Ä§Ä Ä¿Ä¥Ä¼Ä•ÄfÄ ÄCEÄfÄ ÄªÄfÄ Ä,,ÄfÄ

web

Ä¥Ä½Ä¿Ä¥Ä¼Ä ÄfÄ Ä§ÄfÄ€Ä ÄfÄ,Ä½ÄfÄfÄ¼ÄfÄ,Ä¹ÄfÄ,Ä³ÄfÄfÄ¼ÄfÄfÄ%ÄfÄ,Ä~Ä¥Ä Ä,Ä§Ä...Ä§
 ÄfÄ Ä§ÄfÄ Ä ÄfÄ Ä¾ÄfÄ Ä™ÄfÄ€Ä,ÄfÄ Ä“ÄfÄ Ä®Ä!ÄœÄ€Ä!Ä-Ä°ÄfÄ Ä®Ä§Ä^Ä Ä!Ä™Ä~ÄfÄ Ä—ÄfÄ
 ÄfÄ Ä¾ÄfÄ Ä™Ä-

- <http://sosdg.org/~coywolf/lxr/>

+ <http://sosdg.org/~qiyong/lxr/>

ÃŁÃŁ ÃŁ†ÃŁÃŁ€ÃŁ,ÃŸÃŁ...ÃŁ˜ÃŁŸÃŁ”ÃŸÃŁÃŁ ÃŁ˜ÃŁŸÃŁ°ÃŁ;ÃŁ!ÃŁ½ÃŁ”ÃŁÃŁ ÃŁªÃŁ!ÃŁœÃŁ€ÃŁ©ÃŁ«ÃŁ˜ÃŁÃŁ ÃŁ®ÃŁ˜ÃŁŸÃŁÃŁÃŁ,ÃŁ˜ÃŁ
+
ÃŁŸÃŁ”ÃŸÃŁÃŁ ÃŁ˜ÃŁŸÃŁ”ÃŸÃŁŸÃŁ¾ÃŁ’ÃŁÃŁ ÃŁœÃŁ˜ÃŁŸÃŁÃŁÃŁ ÃŁ«ÃŁŸÃŁ˜ÃŁ°ÃŁ©ÃŁ ÃŁ˜ÃŁÃŁ ÃŁ™ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ¾ÃŁÃŁ ÃŁŸÃŁ
+
ÃŁÃŁ ÃŁ†ÃŁÃŁ€ÃŁ,ÃŸÃŁ...ÃŁ˜ÃŁŸÃŁ”ÃŸÃŁÃŁ ÃŁ˜ÃŁŸÃŁ°ÃŁ;ÃŁ!ÃŁ½ÃŁ”ÃŁÃŁ ÃŁªÃŁ!ÃŁœÃŁ€ÃŁ©ÃŁ«ÃŁ˜ÃŁÃŁ ÃŁ®ÃŁ˜ÃŁŸÃŁÃŁ,ÃŁ˜ÃŁ
ÃŁÃŁ ÃŁŸÃŁÃŁ,ÃŁŸÃŁÃŁ€ÃŁ ÃŁÃŁ ÃŁ˜ÃŁÃŁ—ÃŁÃŁ ÃŁ!ÃŁœÃŁ€ÃŁŸÃŁµÃŁ,ÃŁ˜ÃŁŸÃŁÃŁÃŁ ÃŁ®ÃŁŸÃŁ%ÃŁ ÃŁÃŁ ÃŁ®ÃŁµÃŁ,ÃŁ˜ÃŁ
ÃŁÃŁ ÃŁ™”

-
ÃŁÃŁ,ÃŁ«ÃŁÃŁfÃŁ¼ÃŁÃŁfÃŁ ÃŁÃŁfÃŁ«ÃŁ©ÃŁ—ÃŁ˜ÃŁŸÃŁ™ÃŁ°ÃŁÃŁ ÃŁŸÃŁÃŁ,ÃŁ,ÃŁÃŁ ÃŁ“ÃŁÃŁ,ÃŁœÃŁÃŁ ÃŁ˜ŸÃŁ ÃŁœÃŁÃŁ ÃŁ˜ÃŁ
-
ÃŁŸÃŁ•ÃŁ ÃŁ©ÃŁ;ÃŁœÃŁÃŁ,ÃŁ˜ÃŁ˜ÃŁŸÃŁÃŁ!ÃŁ±ÃŁ°ÃŁÃŁ ÃŁ™ÃŁÃŁ,ÃŁ˜ÃŁ˜ÃŁŸÃŁÃŁÃŁ ÃŁ®ÃŁ˜ÃŁfÃŁœÃŁŸÃŁ¾ÃŁœÃŁÃŁ ÃŁ«ÃŁÃŁ ÃŁªÃŁ
-
ÃŁŸÃŁ½ÃŁ¼ÃŁÃŁ,ÃŁ%ÃŁÃŁ ÃŁ˜ŸÃŁ ÃŁ˜ŸÃŁ˜ÃŁ”ÃŁÃŁ ÃŁŸÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁ—ÃŁÃŁ,ÃŁ,ÃŁÃŁ ÃŁ˜ÃŁÃŁ ÃŁªÃŁ˜ÃŁŸÃŁÃŁ!ÃŁ±ÃŁ°ÃŁ!ÃŁ—ÃŁ
+
ÃŁÃŁ,ÃŁ«ÃŁÃŁfÃŁ¼ÃŁÃŁfÃŁ ÃŁÃŁfÃŁ«ÃŁ©ÃŁ—ÃŁ˜ÃŁŸÃŁ™ÃŁ°ÃŁÃŁ ÃŁŸÃŁÃŁ,ÃŁ,ÃŁÃŁ ÃŁ“ÃŁÃŁ,ÃŁœÃŁÃŁ ÃŁ˜ŸÃŁ ÃŁœÃŁÃŁ ÃŁ˜ÃŁ
+
ÃŁŸÃŁ•ÃŁ ÃŁ©ÃŁ;ÃŁœÃŁÃŁ,ÃŁ˜ÃŁ˜ÃŁŸÃŁÃŁ!ÃŁ±ÃŁ°ÃŁÃŁ ÃŁ™ÃŁÃŁ,ÃŁ˜ÃŁ˜ÃŁŸÃŁÃŁÃŁ ÃŁ®ÃŁ˜ÃŁfÃŁœÃŁŸÃŁ¾ÃŁœÃŁÃŁ ÃŁ«ÃŁÃŁ ÃŁªÃŁ
+
ÃŁŸÃŁ½ÃŁ¼ÃŁÃŁ,ÃŁ%ÃŁÃŁ ÃŁ˜ŸÃŁ ÃŁ˜ŸÃŁ˜ÃŁ”ÃŁÃŁ ÃŁŸÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁ—ÃŁÃŁ,ÃŁ,ÃŁÃŁ ÃŁ˜ÃŁÃŁ ÃŁªÃŁ˜ÃŁŸÃŁÃŁ!ÃŁ±ÃŁ°ÃŁ!ÃŁ—ÃŁ

ÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁ—ÃŁÃŁ,ÃŁ,ÃŁÃŁ ÃŁ˜ÃŁÃŁ ÃŁªÃŁ˜ÃŁŸÃŁÃŁ,ÃŁ˜ÃŁ˜ÃŁ—ÃŁ!ÃŁ˜ÃŁ ÃŁÃŁ ÃŁ™ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ®ÃŁÃŁ ÃŁ˜ÃŁÃŁ€ÃŁ
ÃŁ˜ÃŁ—ÃŁ°ÃŁ˜ÃŁ«ÃŁ—ÃŁÃŁ ÃŁ™ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ“ÃŁÃŁ ÃŁ˜ÃŁÃŁ ÃŁ®ÃŁÃŁfÃŁ ÃŁÃŁfÃŁ©ÃŁÃŁfÃŁ³ÃŁÃŁ,ÃŁ!ÃŁÃŁ,ÃŁ˜ÃŁÃŁ,ÃŁ—ÃŁÃŁf
ÃŁÃŁ ÃŁŸÃŁÃŁ ÃŁ™ÃŁÃŁ ÃŁ˜ÃŁÃŁ,ÃŁ%ÃŁÃŁ€ÃŁ ÃŁ©ÃŁ—ÃŁ˜ÃŁŸÃŁ™ÃŁ°ÃŁÃŁfÃŁ—ÃŁÃŁfÃŁ—ÃŁÃŁ,ÃŁ»ÃŁÃŁ,ÃŁ!ÃŁÃŁ ÃŁ®ÃŁ!ÃŁ—ÃŁ
-ÃŁÃŁ ÃŁ†ÃŁÃŁ ÃŁ«ÃŁÃŁ ÃŁ™ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ®ÃŁÃŁ,ÃŁ,ÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁŸÃŁÃŁ ÃŁ™ÃŁÃŁ ÃŁœÃŁÃŁ€ÃŁ ÃŁŸÃŁµÃŁ%
-ÃŁÃŁ ÃŁ!ÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁªÃŁÃŁ ÃŁ,ÃŁµÃŁ»ÃŁ•ÃŁµÃŁ°ÃŁÃŁÃŁ,ÃŁ’(ÃŁ©ÃŁfÃŁ˜ÃŁŸÃŁ˜ÃŁ†ÃŁŸÃŁŸÃŁ,ÃŁÃŁ ÃŁ«)ÃŁŸÃŁ ÃŁ—ÃŁÃŁ,ÃŁŸÃŁ˜ÃŁ
+ÃŁÃŁ ÃŁ†ÃŁÃŁ ÃŁ«ÃŁÃŁ ÃŁ™ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ®ÃŁÃŁ,ÃŁ˜ÃŁ%ÃŁ˜ÃŁÃŁ,ÃŁÃŁ ÃŁŸÃŁÃŁ ÃŁ™ÃŁÃŁ ÃŁœÃŁÃŁ€ÃŁ ÃŁŸÃŁµÃŁ%
+ÃŁÃŁ ÃŁ!ÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁªÃŁÃŁ ÃŁ,ÃŁµÃŁ»ÃŁ•ÃŁµÃŁ°ÃŁÃŁÃŁ,ÃŁ’(ÃŁ©ÃŁfÃŁ˜ÃŁŸÃŁ˜ÃŁ†ÃŁŸÃŁŸÃŁ,ÃŁÃŁ ÃŁ«)ÃŁŸÃŁ ÃŁ—ÃŁÃŁ,ÃŁŸÃŁ˜ÃŁ

ÃŁÃŁ ÃŁ¾ÃŁÃŁ ÃŁŸÃŁÃŁ€ÃŁ ÃŁÃŁ ÃŁŸÃŁÃŁ ÃŁ µÃŁ,ÃŁŸÃŁÃŁ ÃŁœÃŁÃŁ ÃŁÃŁÃŁ ÃŁ!ÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁªÃŁÃŁ ÃŁ,ÃŁÃŁ,ÃŁ,ÃŁÃŁ ÃŁ
ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ†ÃŁÃŁ ÃŁªÃŁÃŁfÃŁ˜ÃŁÃŁfÃŁfÃŁfÃŁ ÃŁÃŁ,ÃŁ˜ÃŁÃŁ€ÃŁ ÃŁ!ÃŁœÃŁ—ÃŁ!ÃŁµÃŁ ÃŁÃŁ ÃŁ«ÃŁŸÃŁ ÃŁ«ÃŁÃŁ,ÃŁ
ÃŁÃŁ ÃŁ!ÃŁÃŁ,ÃŁ,ÃŁÃŁ,ÃŁ%ÃŁÃŁ ÃŁ†ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ†ÃŁÃŁ ÃŁ«ÃŁ©ÃŁ€ÃŁ ÃŁÃŁ ÃŁÃŁÃŁ ÃŁ!ÃŁÃŁ,ÃŁ,ÃŁÃŁ€ÃŁ ÃŁÃŁ ÃŁ ÃŁÃŁ,ÃŁ
@@ -629,7 +635,7 @@ Linux

ÃŁÃŁ,ÃŁ«ÃŁÃŁfÃŁ¼ÃŁÃŁfÃŁ ÃŁÃŁfÃŁ«ÃŁÃŁ,ÃŁŸÃŁÃŁfÃŁŸÃŁfÃŁŸÃŁfÃŁfÃŁ ÃŁÃŁ,ÃŁ˜ÃŁÃŁ€ÃŁ ÃŁ!ÃŁœÃŁ—ÃŁ!ÃŁµÃŁ ÃŁÃŁ ÃŁ«ÃŁŸÃŁ ÃŁ«ÃŁÃŁ,ÃŁ
-ÃŁÃŁfÃŁ†ÃŁÃŁ,ÃŁ!ÃŁÃŁfÃŁ˜ÃŁŸÃŁµÃŁ ÃŁ!ÃŁ ÃŁœ

ÃŁÃŁ ÃŁ“ÃŁÃŁ,ÃŁœÃŁÃŁ ÃŁ«ÃŁÃŁ ÃŁµÃŁÃŁ ÃŁ,ÃŁÃŁ ÃŁ!ÃŁŸÃŁ...ÃŁ˜ÃŁÃŁ ÃŁ!ÃŁÃŁ ÃŁœÃŁÃŁ ÃŁ©ÃŁÃŁ ÃŁ®ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ†ÃŁÃŁ
-ÃŁÃŁfÃŁ³ÃŁÃŁfÃŁ˜ÃŁÃŁ ÃŁ® ChangeLog
ÃŁÃŁ,ÃŁ»ÃŁÃŁ,ÃŁ˜ÃŁÃŁ,ÃŁ˜ÃŁÃŁfÃŁŸÃŁÃŁfÃŁ³ÃŁÃŁ,ÃŁ˜ÃŁÃŁ ÃŁ;ÃŁÃŁ ÃŁ!ÃŁÃŁ ÃŁ ÃŁÃŁ ÃŁÃŁÃŁ ÃŁ•ÃŁÃŁ ÃŁ,-
+ÃŁÃŁfÃŁ³ÃŁÃŁfÃŁ˜ÃŁÃŁ ÃŁ® ChangeLog
ÃŁÃŁ,ÃŁ»ÃŁÃŁ,ÃŁ˜ÃŁÃŁ,ÃŁ˜ÃŁÃŁfÃŁŸÃŁÃŁfÃŁ³ÃŁÃŁ,ÃŁ˜ÃŁ˜ÃŁ!ÃŁ˜ÃŁÃŁ ÃŁ!ÃŁÃŁ ÃŁ ÃŁÃŁ ÃŁ ÃŁÃŁ ÃŁ•ÃŁÃŁ ÃŁ,-
"The Perfect Patch"

<http://www.zip.com.au/~akpm/linux/patches/stuff/tpp.txt>

```
diff --git a/Makefile b/Makefile
index c244a02..c6d545c 100644
--- a/Makefile
+++ b/Makefile
@@ -1,7 +1,7 @@
VERSION = 2
PATCHLEVEL = 6
```

```

SUBLEVEL = 23
-EXTRAVERSION = .1
+EXTRAVERSION = .2
NAME = Arr Matey! A Hairy Bilge Rat!

# *DOCUMENTATION*
diff --git a/block/ll_rw_blk.c b/block/ll_rw_blk.c
index ed39313..026cf24 100644
--- a/block/ll_rw_blk.c
+++ b/block/ll_rw_blk.c
@@ -819,7 +819,6 @@ static int __blk_free_tags(struct blk_queue_tag *bqt)
retval = atomic_dec_and_test(&bqt->refcnt);
if (retval) {
BUG_ON(bqt->busy);
- BUG_ON(!list_empty(&bqt->busy_list));

kfree(bqt->tag_index);
bqt->tag_index = NULL;
@@ -931,7 +930,6 @@ static struct blk_queue_tag * __blk_queue_init_tags(struct request_queue *q,
if (init_tag_map(q, tags, depth))
goto fail;

- INIT_LIST_HEAD(&tags->busy_list);
tags->busy = 0;
atomic_set(&tags->refcnt, 1);
return tags;
@@ -982,6 +980,7 @@ int blk_queue_init_tags(struct request_queue *q, int depth,
*/
q->queue_tags = tags;
q->queue_flags |= (1 << QUEUE_FLAG_QUEUED);
+ INIT_LIST_HEAD(&q->tag_busy_list);
return 0;
fail:
kfree(tags);
@@ -1152,7 +1151,7 @@ int blk_queue_start_tag(struct request_queue *q, struct request *rq)
rq->tag = tag;
bqt->tag_index[tag] = rq;
blkdev_dequeue_request(rq);
- list_add(&rq->queuelist, &bqt->busy_list);
+ list_add(&rq->queuelist, &q->tag_busy_list);
bqt->busy++;
return 0;
}
@@ -1173,11 +1172,10 @@ EXPORT_SYMBOL(blk_queue_start_tag);
**/
void blk_queue_invalidate_tags(struct request_queue *q)
{
- struct blk_queue_tag *bqt = q->queue_tags;
struct list_head *tmp, *n;
struct request *rq;

```

```

- list_for_each_safe(tmp, n, &bqt->busy_list) {
+ list_for_each_safe(tmp, n, &q->tag_busy_list) {
rq = list_entry_rq(tmp);

if (rq->tag == -1) {
diff --git a/fs/locks.c b/fs/locks.c
index c795eaa..494f250 100644
--- a/fs/locks.c
+++ b/fs/locks.c
@@ -694,11 +694,20 @@ EXPORT_SYMBOL(posix_test_lock);
* Note: the above assumption may not be true when handling lock requests
* from a broken NFS client. But broken NFS clients have a lot more to
* worry about than proper deadlock detection anyway... --okir
+ *
+ * However, the failure of this assumption (also possible in the case of
+ * multiple tasks sharing the same open file table) also means there's no
+ * guarantee that the loop below will terminate. As a hack, we give up
+ * after a few iterations.
*/
+
+#define MAX_DEADLK_ITERATIONS 10
+
static int posix_locks_deadlock(struct file_lock *caller_fl,
struct file_lock *block_fl)
{
struct list_head *tmp;
+ int i = 0;

next_task:
if (posix_same_owner(caller_fl, block_fl))
@@ -706,6 +715,8 @@ next_task:
list_for_each(tmp, &blocked_list) {
struct file_lock *fl = list_entry(tmp, struct file_lock, fl_link);
if (posix_same_owner(fl, block_fl)) {
+ if (i++ > MAX_DEADLK_ITERATIONS)
+ return 0;
fl = fl->fl_next;
block_fl = fl;
goto next_task;
diff --git a/fs/proc/array.c b/fs/proc/array.c
index ee4814d..20d7ae4 100644
--- a/fs/proc/array.c
+++ b/fs/proc/array.c
@@ -351,7 +351,8 @@ static cputime_t task_utime(struct task_struct *p)
}
utime = (clock_t)temp;

- return clock_t_to_cputime(utime);
+ p->prev_utime = max(p->prev_utime, clock_t_to_cputime(utime));
+ return p->prev_utime;
}

```

```
static cputime_t task_stime(struct task_struct *p)
@@ -366,7 +367,8 @@ static cputime_t task_stime(struct task_struct *p)
stime = nsec_to_clock_t(p->se.sum_exec_runtime) -
cputime_to_clock_t(task_utime(p));
```

```
- return clock_t_to_cputime(stime);
+ p->prev_stime = max(p->prev_stime, clock_t_to_cputime(stime));
+ return p->prev_stime;
}
#endif
```

```
diff --git a/fs/splice.c b/fs/splice.c
index e95a362..02c39ae 100644
```

```
--- a/fs/splice.c
+++ b/fs/splice.c
@@ -1390,10 +1390,10 @@ static int pipe_to_user(struct pipe_inode_info *pipe, struct pipe_buffer *buf,
if (copy_to_user(sd->u.userptr, src + buf->offset, sd->len))
ret = -EFAULT;
```

```
+ buf->ops->unmap(pipe, buf, src);
out:
if (ret > 0)
sd->u.userptr += ret;
- buf->ops->unmap(pipe, buf, src);
return ret;
}
```

```
diff --git a/include/linux/blkdev.h b/include/linux/blkdev.h
index b126c6f..d26bbb0 100644
```

```
--- a/include/linux/blkdev.h
+++ b/include/linux/blkdev.h
@@ -356,7 +356,6 @@ enum blk_queue_state {
struct blk_queue_tag {
struct request **tag_index; /* map of busy tags */
unsigned long *tag_map; /* bit map of free/busy tags */
- struct list_head busy_list; /* fifo list of busy tags */
int busy; /* current depth */
int max_depth; /* what we will send to device */
int real_max_depth; /* what the array can hold */
@@ -451,6 +450,7 @@ struct request_queue
unsigned int dma_alignment;
```

```
struct blk_queue_tag *queue_tags;
+ struct list_head tag_busy_list;
```

```
unsigned int nr_sorted;
unsigned int in_flight;
```

```
diff --git a/include/linux/sched.h b/include/linux/sched.h
index 313c6b6..f509fbd 100644
--- a/include/linux/sched.h
```

```

+++ b/include/linux/sched.h
@@ -1022,6 +1022,7 @@ struct task_struct {

unsigned int rt_priority;
cputime_t utime, stime;
+ cputime_t prev_utime, prev_stime;
unsigned long nvcsw, nivcsw; /* context switch counts */
struct timespec start_time; /* monotonic time */
struct timespec real_start_time; /* boot based time */
diff --git a/kernel/fork.c b/kernel/fork.c
index 33f12f4..f299d45 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1045,6 +1045,8 @@ static struct task_struct *copy_process(unsigned long clone_flags,

p->utime = cputime_zero;
p->stime = cputime_zero;
+ p->prev_utime = cputime_zero;
+ p->prev_stime = cputime_zero;

#ifdef CONFIG_TASK_XACCT
p->rchar = 0; /* I/O counter: bytes read */
diff --git a/kernel/futex_compat.c b/kernel/futex_compat.c
index 2c2e295..f938c23 100644
--- a/kernel/futex_compat.c
+++ b/kernel/futex_compat.c
@@ -29,6 +29,15 @@ fetch_robust_entry(compat_uptr_t *uentry, struct robust_list __user **entry,
return 0;
}

+static void __user *futex_uaddr(struct robust_list *entry,
+ compat_long_t futex_offset)
+{
+ compat_uptr_t base = ptr_to_compat(entry);
+ void __user *uaddr = compat_ptr(base + futex_offset);
+
+ return uaddr;
+}
+
+/*
+ * Walk curr->robust_list (very carefully, it's a userspace list!)
+ * and mark any locks found there dead, and notify any waiters.
@@ -75,11 +84,13 @@ void compat_exit_robust_list(struct task_struct *curr)
+ * A pending lock might already be on the list, so
+ * dont process it twice:
+ */
- if (entry != pending)
- if (handle_futex_death((void __user *)entry + futex_offset,
- curr, pi))
- return;
+ if (entry != pending) {

```

```

+ void __user *uaddr = futex_uaddr(entry,
+ futex_offset);

+ if (handle_futex_death(uaddr, curr, pi))
+ return;
+ }
if (rc)
return;
uentry = next_uentry;
@@ -93,9 +104,11 @@ void compat_exit_robust_list(struct task_struct *curr)

```

```

cond_resched();
}
- if (pending)
- handle_futex_death((void __user *)pending + futex_offset,
- curr, pip);
+ if (pending) {
+ void __user *uaddr = futex_uaddr(pending, futex_offset);
+
+ handle_futex_death(uaddr, curr, pip);
+ }
}

```

```

asmlinkage long
diff --git a/kernel/lockdep.c b/kernel/lockdep.c
index 734da57..42ae4a5 100644
--- a/kernel/lockdep.c
+++ b/kernel/lockdep.c
@@ -1521,7 +1521,7 @@ cache_hit:
}

```

```

static int validate_chain(struct task_struct *curr, struct lockdep_map *lock,
- struct held_lock *hlock, int chain_head)
+ struct held_lock *hlock, int chain_head, u64 chain_key)
{
/*
* Trylock needs to maintain the stack of held locks, but it
@@ -1534,7 +1534,7 @@ static int validate_chain(struct task_struct *curr, struct lockdep_map *lock,
* graph_lock for us)
*/
if (!hlock->trylock && (hlock->check == 2) &&
- lookup_chain_cache(curr->curr_chain_key, hlock->class)) {
+ lookup_chain_cache(chain_key, hlock->class)) {
/*
* Check whether last held lock:
*
@@ -1576,7 +1576,7 @@ static int validate_chain(struct task_struct *curr, struct lockdep_map *lock,
#else
static inline int validate_chain(struct task_struct *curr,
struct lockdep_map *lock, struct held_lock *hlock,
- int chain_head)

```

```

+ int chain_head, u64 chain_key)
{
return 1;
}
@@ -2450,11 +2450,11 @@ static int __lock_acquire(struct lockdep_map *lock, unsigned int subclass,
chain_head = 1;
}
chain_key = iterate_chain_key(chain_key, id);
- curr->curr_chain_key = chain_key;

- if (!validate_chain(curr, lock, hlock, chain_head))
+ if (!validate_chain(curr, lock, hlock, chain_head, chain_key))
return 0;

+ curr->curr_chain_key = chain_key;
curr->lockdep_depth++;
check_chain_key(curr);
#ifdef CONFIG_DEBUG_LOCKDEP
diff --git a/kernel/params.c b/kernel/params.c
index 4e57732..5e5651f 100644
--- a/kernel/params.c
+++ b/kernel/params.c
@@ -595,13 +595,16 @@ static void __init param_sysfs_builtin(void)

for (i=0; i < __stop__param - __start__param; i++) {
char *dot;
+ size_t max_name_len;

kp = &__start__param[i];
+ max_name_len =
+ min_t(size_t, MAX_KBUILD_MODNAME, strlen(kp->name));

- /* We do not handle args without periods. */
- dot = memchr(kp->name, '.', MAX_KBUILD_MODNAME);
+ dot = memchr(kp->name, '.', max_name_len);
if (!dot) {
- DEBUGP("couldn't find period in %s\n", kp->name);
+ DEBUGP("couldn't find period in first %d characters "
+ "of %s\n", MAX_KBUILD_MODNAME, kp->name);
continue;
}
name_len = dot - kp->name;
diff --git a/kernel/softlockup.c b/kernel/softlockup.c
index 708d488..e557c44 100644
--- a/kernel/softlockup.c
+++ b/kernel/softlockup.c
@@ -80,10 +80,11 @@ void softlockup_tick(void)
print_timestamp = per_cpu(print_timestamp, this_cpu);

/* report at most once a second */
- if (print_timestamp < (touch_timestamp + 1) ||

```

```

- did_panic ||
- !per_cpu(watchdog_task, this_cpu))
+ if ((print_timestamp >= touch_timestamp &&
+ print_timestamp < (touch_timestamp + 1)) ||
+ did_panic || !per_cpu(watchdog_task, this_cpu)) {
return;
+ }

/* do not print during early bootup: */
if (unlikely(system_state != SYSTEM_RUNNING)) {
diff --git a/mm/filemap.c b/mm/filemap.c
index 15c8413..14ca63f 100644
--- a/mm/filemap.c
+++ b/mm/filemap.c
@@ -1312,7 +1312,7 @@ int filemap_fault(struct vm_area_struct *vma, struct vm_fault *vmf)

size = (i_size_read(inode) + PAGE_CACHE_SIZE - 1) >> PAGE_CACHE_SHIFT;
if (vmf->pgoff >= size)
- goto outside_data_content;
+ return VM_FAULT_SIGBUS;

/* If we don't want any read-ahead, don't bother */
if (VM_RandomReadHint(vma))
@@ -1389,7 +1389,7 @@ retry_find:
if (unlikely(vmf->pgoff >= size)) {
unlock_page(page);
page_cache_release(page);
- goto outside_data_content;
+ return VM_FAULT_SIGBUS;
}

/*
@@ -1400,15 +1400,6 @@ retry_find:
vmf->page = page;
return ret | VM_FAULT_LOCKED;

-outside_data_content:
- /*
- * An external ptracer can access pages that normally aren't
- * accessible..
- */
- if (vma->vm_mm == current->mm)
- return VM_FAULT_SIGBUS;
-
- /* Fall through to the non-read-ahead case */
no_cached_page:
/*
* We're only likely to ever get here if MADV_RANDOM is in
diff --git a/mm/page-writeback.c b/mm/page-writeback.c
index 4472036..97ddc58 100644
--- a/mm/page-writeback.c

```

```

+++ b/mm/page-writeback.c
@@ -672,8 +672,10 @@ retry:

ret = (*writepage)(page, wbc, data);

- if (unlikely(ret == AOP_WRITEPAGE_ACTIVATE))
+ if (unlikely(ret == AOP_WRITEPAGE_ACTIVATE)) {
unlock_page(page);
+ ret = 0;
+ }
if (ret || (--(wbc->nr_to_write) <= 0))
done = 1;
if (wbc->nonblocking && bdi_write_congested(bdi)) {
diff --git a/mm/shmem.c b/mm/shmem.c
index fcd19d3..95558e4 100644
--- a/mm/shmem.c
+++ b/mm/shmem.c
@@ -916,6 +916,21 @@ static int shmem_writepage(struct page *page, struct writeback_control *wbc)
struct inode *inode;

BUG_ON(!PageLocked(page));
+ /*
+ * shmem_backing_dev_info's capabilities prevent regular writeback or
+ * sync from ever calling shmem_writepage; but a stacking filesystem
+ * may use the ->writepage of its underlying filesystem, in which case
+ * we want to do nothing when that underlying filesystem is tmpfs
+ * (writing out to swap is useful as a response to memory pressure, but
+ * of no use to stabilize the data) - just redirty the page, unlock it
+ * and claim success in this case. AOP_WRITEPAGE_ACTIVATE, and the
+ * page_mapped check below, must be avoided unless we're in reclaim.
+ */
+ if (!wbc->for_reclaim) {
+ set_page_dirty(page);
+ unlock_page(page);
+ return 0;
+ }
BUG_ON(page_mapped(page));

mapping = page->mapping;
diff --git a/mm/slub.c b/mm/slub.c
index addb20a..c1f2fda 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -1501,28 +1501,8 @@ new_slab:
page = new_slab(s, gfpflags, node);
if (page) {
cpu = smp_processor_id();
- if (s->cpu_slab[cpu]) {
- /*
- * Someone else populated the cpu_slab while we
- * enabled interrupts, or we have gotten scheduled

```

```
- * on another cpu. The page may not be on the
- * requested node even if __GFP_THISNODE was
- * specified. So we need to recheck.
- */
- if (node == -1 ||
- page_to_nid(s->cpu_slab[cpu]) == node) {
- /*
- * Current cpuslab is acceptable and we
- * want the current one since its cache hot
- */
- discard_slab(s, page);
- page = s->cpu_slab[cpu];
- slab_lock(page);
- goto load_freelist;
- }
- /* New slab does not fit our expectations */
+ if (s->cpu_slab[cpu])
flush_slab(s, s->cpu_slab[cpu], cpu);
- }
slab_lock(page);
SetSlabFrozen(page);
s->cpu_slab[cpu] = page;
-
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>