È disponibile un nuovo kernel 2.6.30-rc3 terzo rilascio da http://www.kernel.org/

Ricordo che questo kernel è usabile con attenzione ma che manca di tutte le patch e moduli 3rdparty tipiche dei kernel Mandriva non essendo quello ufficiale di Mandriva.

driver synchronization: make scsi_wait_scan more advanced

There is currently only one way for userspace to say "wait for my storage device to get ready for the modules I just loaded": to load the scsi_wait_scan module. Expectations of userspace are that once this module is loaded, all the (storage) devices for which the drivers were loaded before the module load are present.

Now, there are some issues with the implementation, and the async stuff got caught in the middle of this: The existing code only waits for the scsy async probing to finish, but it did not take into account at all that probing might not have begun yet.
(Russell ran into this problem on his computer and the fix works for him)

This patch fixes this more thoroughly than the previous "fix", which had some bad side effects (namely, for kernel code that wanted to wait for the scsi scan it would also do an async sync, which would deadlock if you did it from async context already.. there's a report about that on lkml):
The patch makes the module first wait for all device driver probes, and then it will wait for the scsi parallel scan to finish.

commit 5dd559f020c98a2a4b3e063f09c0e4bc771ed838

Trivial: fix a typo in slow-work.h

Fix a comment typo in slow-work.h

```
    ...a trivial mistake, but it will mess up kerneldoc if nothing else.

 commit 5028eaa97dd1dab9cd7c30c4d38f71c708ca64bc

    PERCPU: Collect the DECLARE/DEFINE declarations together

    Collect the DECLARE/DEFINE declarations together in linux/percpu-defs.h so
    that they're in one place, and give them descriptive comments, particularly
    the SHARED_ALIGNED variant.

    It would be nice to collect these in linux/percpu.h, but that's not possible
    without sorting out the severe #include recursion between the x86 arch headers
    and the general headers (and possibly other arches too).

 commit 9b8de7479d0dbab1ed98b5b015d44232c9d3d08e

    FRV: Fix the section attribute on UP DECLARE_PER_CPU()

    In non-SMP mode, the variable section attribute specified by DECLARE_PER_CPU()
    does not agree with that specified by DEFINE_PER_CPU().  This means that
    architectures that have a small data section references relative to a base
    register may throw up linkage errors due to too great a displacement between
    where the base register points and the per-CPU variable.

    On FRV, the .h declaration says that the variable is in the .sdata section, but
    the .c definition says it's actually in the .data section.  The linker throws
    up the following errors:

    kernel/built-in.o: In function `release_task':
    kernel/exit.c:78: relocation truncated to fit: R_FRV_GPREL12 against symbol
`per_cpu__process_counts' defined in .data section in kernel/built-in.o
    kernel/exit.c:78: relocation truncated to fit: R_FRV_GPREL12 against symbol
`per_cpu__process_counts' defined in .data section in kernel/built-in.o

    To fix this, DECLARE_PER_CPU() should simply apply the same section attribute
    as does DEFINE_PER_CPU().  However, this is made slightly more complex by
    virtue of the fact that there are several variants on DEFINE, so these need to
    be matched by variants on DECLARE.
```

Potete scaricarlo da qui, , al solito tasto destro salva destinazione con nome.

Ecco il log dei fix.

ciao.

grissino.