

Appendix: Submitted patches

To illustrate the types of issues that we have found with the VAMPYR tool, we present a selection of the patches that we have submitted to the L4/FIASCO and Linux maintainers.

P1: On L4/FIASCO, we propose the following change to fix the compilation issue that was caused by a missing template parameter and is described in Section 5.2:

```
+++ b/src/kern/ux/main-ux.cpp
-17,7 +17,7
{
  if (!Per_cpu_data_alloc::alloc(_cpu))
  {
-   extern Spin_lock _tramp_mp_spinlock;
+   extern Spin_lock<Mword> _tramp_mp_spinlock;
    printf("CPU allocation failed for CPU%u, disabling CPU.\n", _cpu);
    _tramp_mp_spinlock.clear();
    while (1)
```

P2: The icside driver fails to compile if the driver is compiled without DMA support, which is a valid configuration according to the constraints specified in KCONFIG. This configuration references DMA operations unconditionally since at least commit 5e37bd (dated from Apr 26, 2008). We propose the addition of additional #ifdef statements¹:

```
+++ b/drivers/ide/icside.c
-456,5 +456,7 err_free:
static const struct ide_port_info icside_v6_port_info __initdata = {
    .init_dma          = icside_dma_off_init,
    .port_ops         = &icside_v6_no_dma_port_ops,
+ifdef CONFIG_BLK_DEV_IDEDMA_ICs
    .dma_ops          = &icside_v6_dma_ops,
+endif
    .host_flags       = IDE_HFLAG_SERIALIZE | IDE_HFLAG_MMIO,
-517,7 +519,9 icside_register_v6(struct icside_state *state, struct expansion_card *ec)
    ecard_set_drvdata(ec, state);

+ifdef CONFIG_BLK_DEV_IDEDMA_ICs
    if (ec->dma != NO_DMA && !request_dma(ec->dma, DRV_NAME)) {
        d.init_dma = icside_dma_init;
        d.port_ops = &icside_v6_port_ops;
    } else
+endif
        d.dma_ops = NULL;
```

P3: The icside driver causes also a warning during the compilation, when the DMA support is enabled. Here the first argument to printk, the format string, requires the last argument to be an int. However, the last argument is an unsigned long. This bug was (probably) introduced in commit 5bfb151f, authored June 2009. We therefore propose the following change²:

```
+++ b/drivers/ide/icside.c
-272,5 +272,5 static void icside_set_dma_mode(...)
    ide_set_drvdata(drive, (void *)cycle_time);

-   printk("%s: %s selected (peak %dMB/s)\n", drive->name,
+   printk("%s: %s selected (peak %luMB/s)\n", drive->name,
        ide_xfer_verbose(xfer_mode),
        2000 / (unsigned long)ide_get_drvdata(drive));
```

P4: When compiling the lp5521 LED driver in Linux v3.2, we notice the following gcc warning:

```
drivers/leds/leds-lp5521.c:741: warning: 'buf' may be used uninitialized in this function
```

An inspection of the code reveals an code path that indeed leaves the variable buf uninitialized. This bug was (probably) introduced in commit b3c49c, authored October 2011. We therefore propose the following change³:

```
+++ b/drivers/leds/leds-lp5521.c
-785,7 +785,7 static int __devinit lp5521_probe(...)
    * LP5521_REG_ENABLE register will not have any effect
    */
    ret = lp5521_read(client, LP5521_REG_R_CURRENT, &buf);
-   if (buf != LP5521_REG_R_CURR_DEFAULT) {
+   if (ret == -EIO || buf != LP5521_REG_R_CURR_DEFAULT) {
        dev_err(&client->dev, "error in resetting chip\n");
        goto fail2;
    }
}
```

¹<https://lkml.org/lkml/2012/5/31/163>

²<https://lkml.org/lkml/2012/6/5/191>

³<https://lkml.org/lkml/2012/5/21/262>

P5: The mbx framebuffer driver for the Intel 2700G LCD controller provides helper functions for debugging purposes that are only available of the KCONFIG option FB_MBX_DEBUG is enabled. However, the necessary function prototypes are missing at compilation time when the feature is enabled in KCONFIG. Instead of introducing a header file, we propose a less intrusive solution⁴ to the issue, which has been present since the introduction of the driver in July 2006:

```
+++ b/drivers/video/mbx/mbxfb.c
-878,4 +878,7 static int mbxfb_resume(...)
#ifdef CONFIG_FB_MBX_DEBUG
#define mbxfb_debugfs_init(x) do {} while(0)
#define mbxfb_debugfs_remove(x) do {} while(0)
+else
+void mbxfb_debugfs_init(struct fb_info *fbi);
+void mbxfb_debugfs_remove(struct fb_info *fbi);
#endif
```

P6: The GPIO interface driver for ARM v6 based Qualcomm MSM chips can be compiled in configurations that lead to the following compilation failure:

```
drivers/gpio/gpio-msm-v1.c: In function 'msm_init_gpio':
drivers/gpio/gpio-msm-v1.c:629: error: 'INT_GPIO_GROUP1' undeclared
drivers/gpio/gpio-msm-v1.c:630: error: 'INT_GPIO_GROUP2' undeclared
```

These identifiers are only provided for three specific MSM systems, which are all identified by corresponding KCONFIG features. On all other ARM systems the identifiers are not declared. Our proposed change⁵ for this problem, which lasts in Linux since September 2010 (commit 2783cc26), by adding the missing KCONFIG constraints:

```
+++ b/drivers/gpio/Kconfig
-136,7 +136,7 config GPIO_MPC8XXX

config GPIO_MSM_V1
    tristate "Qualcomm MSM GPIO v1"
- depends on GPIOLIB && ARCH_MSM
+ depends on GPIOLIB && ARCH_MSM && (ARCH_MSM7X00A || ARCH_MSM7X30 || ARCH_QSD8X50)
    help
    Say yes here to support the GPIO interface on
    ARM v6 based Qualcomm MSM chips.
```

P7: In Busybox the compiler warns, that some functions have format-string security problems. In case of coreutils/stat.c it is confirmed as a bug⁶. Therefore we sent in a patch and this was the result⁷ to be integrated in Busybox.

```
+++ b/coreutils/stat.c
-442,7 +442,7 static bool do_statfs(const char *filename, const
char *format)
    : getfilecon(filename, &scontext)
    ) < 0
    ) {
-         bb_perror_msg(filename);
+         bb_simple_perror_msg(filename);
        return 0;
    }
}
-555,7 +555,7 static bool do_stat(const char *filename, const
char *format)
    : getfilecon(filename, &scontext)
    ) < 0
    ) {
-         bb_perror_msg(filename);
+         bb_simple_perror_msg(filename);
        return 0;
    }
}
+++ b/e2fsprogs/old_e2fsprogs/lsattr.c
-93,7 +93,7 static int lsattr_dir_proc(const char *dir_name,
struct dirent *de,
    path = concat_path_file(dir_name, de->d_name);

    if (lstat(path, &st) == -1)
-         bb_perror_msg(path);
+         bb_simple_perror_msg(path);
    else {
        if (de->d_name[0] != '.' || (flags & OPT_ALL)) {
            list_attributes(path);
```

⁴<https://lkml.org/lkml/2012/6/5/467>

⁵<https://lkml.org/lkml/2012/5/31/181>

⁶<http://lists.busybox.net/pipermail/busybox/2012-September/078360.html>

⁷<http://lists.busybox.net/pipermail/busybox/2012-September/078473.html>

Appendix: Coverage data for all architectures

Architecture	#files	Total kLOC	in #ifdef blocks	# variation		allyes CC _S	allyes CC _N	VAMPYR CC _S	VAMPYR CC _N
				points (dead/undead rate)					
alpha	9,182	7,535	3.3%	28,630	(34%)	55.4%	83.2%	60%	90.2%
hardware	6,527	5,649.2	2.5%	20,414	(40%)	48.3%	80.3%	52.3%	87%
drivers/	5,822	5,131.6	2.2%	17,786	(41%)	47.2%	80.2%	51%	86.6%
arch/	142	42.5	7.5%	302	(2%)	54.3%	55.4%	66.2%	67.6%
sound/	563	475	4.2%	2,326	(35%)	55.6%	86%	60.7%	93.9%
software	2,655	1,885.8	6%	8,216	(18%)	72.9%	88.3%	79.2%	95.9%
kernel/	1,251	1,053.1	6.1%	4,292	(27%)	62.7%	84.4%	69.3%	93.4%
net/	1,022	655.4	6.1%	3,158	(7%)	87.2%	93.6%	92.8%	99.7%
rest	382	177.3	5.4%	766	(14%)	71.5%	82.9%	78.5%	90.7%
arm	11,593	8,567.6	4.6%	33,356	(18%)	49.2%	59.9%	69.4%	84.4%
hardware	8,883	6,629.2	3.9%	25,140	(20%)	40.8%	51.2%	63.9%	80.1%
drivers/	6,377	5,606.1	3.1%	17,786	(26%)	49.6%	67.2%	65.9%	89.3%
arch/	1,818	494.2	12.6%	5,028	(3%)	4.2%	4.4%	48.6%	49.8%
sound/	688	528.9	5.2%	2,326	(12%)	53.4%	60.9%	82.2%	93.6%
software	2,710	1,938.3	6.9%	8,216	(11%)	74.8%	83.6%	86.3%	96.3%
kernel/	1,293	1,101.5	7.5%	4,292	(14%)	66.2%	76.9%	81.5%	94.4%
net/	1,029	657.1	6.1%	3,158	(6%)	87%	92.6%	93.7%	99.7%
rest	388	179.7	5.7%	766	(10%)	73.2%	80.9%	82.9%	91.4%
avr32	6,823	4,923.3	4.1%	28,578	(49%)	40.9%	79.5%	45.2%	87.9%
hardware	4,184	3,053.3	2.8%	20,362	(61%)	27.8%	71.6%	31.3%	80.6%
drivers/	3,808	2,785.8	2.5%	17,786	(62%)	27.9%	73%	31.1%	81.3%
arch/	85	16.1	14.3%	250	(1%)	28.8%	29%	59.6%	60.1%
sound/	291	251.4	5.5%	2,326	(63%)	27.2%	72.6%	30.1%	80.4%
software	2,639	1,870.1	6.1%	8,216	(18%)	73.2%	88.8%	79.6%	96.5%
kernel/	1,258	1,054.4	6.1%	4,292	(26%)	64.2%	85.6%	71%	94.8%
net/	1,000	638.8	6.1%	3,158	(8%)	85.8%	93.6%	91.4%	99.7%
rest	381	176.8	5.4%	766	(14%)	71.8%	82.8%	78.5%	90.5%
blackfin	6,868	4,968.9	4.6%	30,473	(45%)	39.2%	70.5%	46.1%	82.8%
hardware	4,241	3,091.2	3.5%	22,257	(55%)	26.7%	59.8%	32.8%	73.4%
drivers/	3,801	2,792.2	2.6%	17,786	(61%)	28.1%	71.6%	32.1%	81.9%
arch/	130	42	47.1%	2,145	(6%)	12.4%	13.1%	38.5%	40.7%
sound/	310	257	5.5%	2,326	(60%)	29.3%	72.8%	32.8%	81.6%
software	2,627	1,877.7	6.5%	8,216	(15%)	73.1%	85.6%	82.2%	96.2%
kernel/	1,249	1,062.6	7%	4,292	(20%)	64.2%	79.7%	76%	94.2%
net/	996	637.9	6.1%	3,158	(8%)	85.7%	93.5%	91.4%	99.7%
rest	382	177.1	5.4%	766	(14%)	71.1%	82%	79.2%	91.2%
cris	9,747	7,792.2	2.4%	28,893	(38%)	26.3%	42.4%	35.6%	57.6%
hardware	7,166	5,964	1.4%	20,677	(45%)	9.1%	16.5%	19.9%	36%
drivers/	6,244	5,344.8	1.4%	17,786	(47%)	10%	18.7%	21.8%	40.6%
arch/	118	37	8.2%	565	(7%)	18.2%	19.3%	41.2%	43.9%
sound/	804	582.2	1.7%	2,326	(43%)	0%	0%	0%	0%
software	2,581	1,828.2	5.6%	8,216	(22%)	69.5%	88.4%	75.3%	95.8%
kernel/	1,228	1,021.6	5.3%	4,292	(31%)	59.3%	85.5%	64.9%	93.7%
net/	987	633.6	6.1%	3,158	(10%)	83.7%	92.7%	89.9%	99.5%
rest	366	173	5.3%	766	(17%)	68%	82.1%	74%	89.4%
frv	8,746	7,303.7	3.5%	28,562	(34%)	54.8%	82.6%	59.6%	90%
hardware	6,111	5,440.9	2.8%	20,346	(40%)	47.9%	79.1%	52.4%	86.5%
drivers/	5,579	5,005	2.6%	17,786	(40%)	47.3%	79.2%	51.5%	86.3%
arch/	73	18.1	7.7%	234	(3%)	45.3%	46.5%	58.1%	59.7%
sound/	459	417.8	5.1%	2,326	(37%)	52.8%	83.8%	58.2%	92.4%

Architecture	#files	Total kLOC	in #ifdef blocks	# variation points (dead/undead rate)	allyes CC _S	allyes CC _N	VAMPYR CC _S	VAMPYR CC _N	
software	2,635	1,862.8	5.6%	8,216	(20%)	71.8%	89.2%	77.7%	96.5%
kernel/	1,240	1,034.5	5.5%	4,292	(30%)	61.1%	86%	67.4%	94.9%
net/	1,022	654.6	6%	3,158	(8%)	86.6%	93.7%	92.1%	99.7%
rest	373	173.6	5.4%	766	(16%)	70.1%	83.2%	75.7%	89.9%
h8300	8,530	6,335.2	2.3%	28,482	(48%)	24%	45.7%	26.1%	49.8%
hardware	5,991	4,538.4	1.1%	20,266	(58%)	6%	14.1%	6.9%	16.1%
drivers/	5,143	3,951.6	1%	17,786	(60%)	6.7%	16.4%	7.5%	18.5%
arch/	44	5.7	7.9%	154	(4%)	21.4%	22.3%	38.3%	39.2%
sound/	804	581.1	1.5%	2,326	(45%)	0%	0%	0%	0%
software	2,539	1,796.8	5.3%	8,216	(24%)	68.2%	89.1%	73.5%	96.1%
kernel/	1,193	992.4	4.8%	4,292	(35%)	56.6%	86%	61.7%	93.9%
net/	985	633.1	6.1%	3,158	(10%)	84.3%	93.6%	89.7%	99.5%
rest	361	171.3	5.2%	766	(19%)	67%	82.8%	72.6%	89.8%
hexagon	5,865	4,133	4.3%	28,387	(56%)	34.8%	77.9%	38.3%	85.7%
hardware	3,209	2,232	2.4%	20,171	(72%)	18.4%	65.8%	20.5%	73.2%
drivers/	2,914	2,001.9	2.1%	17,786	(73%)	17.7%	64.4%	19.9%	72.4%
arch/	44	6.4	1.6%	59	(5%)	64.4%	67.9%	64.4%	67.9%
sound/	251	223.7	5%	2,326	(69%)	22.8%	74.8%	24.2%	79.3%
software	2,656	1,901	6.5%	8,216	(15%)	75%	87.7%	81.9%	95.8%
kernel/	1,277	1,085.1	6.8%	4,292	(21%)	67.6%	83.8%	75.3%	93.4%
net/	995	638	6.2%	3,158	(8%)	85.7%	93.5%	91.3%	99.7%
rest	384	177.9	5.5%	766	(12%)	73.2%	83.2%	80.7%	91.5%
ia64	8,995	7,443.3	4%	28,933	(31%)	57.8%	83%	63.7%	91.6%
hardware	6,329	5,545.6	3.2%	20,717	(37%)	50.9%	80.8%	56.1%	89.1%
drivers/	5,678	5,049.4	2.8%	17,786	(38%)	50.4%	81.6%	55.3%	89.6%
arch/	192	77.8	16.2%	605	(3%)	57%	57.8%	69.1%	70.3%
sound/	459	418.4	5.2%	2,326	(37%)	53.1%	83.9%	58.7%	92.7%
software	2,666	1,897.6	6.5%	8,216	(14%)	75.1%	87.1%	82.9%	96.1%
kernel/	1,256	1,062.3	6.9%	4,292	(21%)	66.2%	82.5%	75.5%	94.3%
net/	1,025	656.5	6.1%	3,158	(7%)	87.6%	93.6%	93.3%	99.7%
rest	385	178.9	5.6%	766	(10%)	73.9%	82%	81.1%	89.8%
m32r	5,810	4,043.5	4.1%	28,730	(56%)	33.4%	76.4%	37.5%	85.8%
hardware	3,191	2,183.2	2.5%	20,514	(72%)	17.9%	63.5%	21%	74.3%
drivers/	2,867	1,939.7	2.1%	17,786	(74%)	17.1%	64.5%	19.4%	73.4%
arch/	59	13.2	24.7%	402	(11%)	22.9%	25.8%	66.4%	75%
sound/	265	230.4	4.9%	2,326	(69%)	23.5%	75.2%	24.9%	79.6%
software	2,619	1,860.3	6.1%	8,216	(18%)	71.9%	87.4%	78.7%	95.7%
kernel/	1,245	1,045.5	6.1%	4,292	(26%)	62.2%	83%	69.8%	93.3%
net/	992	637.9	6.2%	3,158	(9%)	85.4%	93.5%	91%	99.7%
rest	382	177	5.3%	766	(14%)	70.8%	82.1%	78.1%	90.4%
m68k	7,059	4,998.1	3.7%	28,936	(49%)	33.1%	65.1%	43.3%	85.3%
hardware	4,441	3,149.9	2.5%	20,720	(61%)	18.2%	46.5%	29.9%	76.4%
drivers/	4,111	2,992	2.4%	17,786	(59%)	20.4%	49.2%	33.3%	80.4%
arch/	237	77.4	6.7%	608	(3%)	23.4%	24%	43.8%	44.9%
sound/	93	80.5	2.3%	2,326	(93%)	0.4%	5.8%	0.5%	7.8%
software	2,618	1,848.2	5.7%	8,216	(20%)	70.6%	88%	77.2%	96.2%
kernel/	1,244	1,034.7	5.5%	4,292	(29%)	59.9%	84.1%	67.2%	94.3%
net/	993	637.4	6.1%	3,158	(9%)	85.2%	93.4%	90.9%	99.7%
rest	381	176.1	5.2%	766	(15%)	70.4%	82.4%	76.8%	89.9%
microblaze	8,746	7,231.2	3.4%	28,531	(36%)	52.6%	82.4%	57.2%	89.7%
hardware	6,055	5,320.4	2.4%	20,315	(45%)	43.8%	79%	47.5%	85.8%
drivers/	5,536	4,896.4	2.2%	17,786	(45%)	43.7%	79%	47.3%	85.4%

Architecture	#files	Total kLOC	in #ifdef blocks	# variation points (dead/undead rate)	allyes CC _S	allyes CC _N	VAMPYR CC _S	VAMPYR CC _N	
arch/	66	13.3	17.4%	203	(2%)	47.3%	47.2%	63.1%	63.3%
sound/	453	410.8	4.1%	2,326	(49%)	43.6%	84.9%	47.9%	93.1%
software	2,691	1,910.8	6.2%	8,216	(16%)	74.3%	88%	81.2%	96.2%
kernel/	1,283	1,077.3	6.4%	4,292	(23%)	65.6%	84.2%	73.4%	94.4%
net/	1,025	655	5.9%	3,158	(7%)	86.6%	93.6%	92.2%	99.7%
rest	383	178.4	5.5%	766	(12%)	72.2%	82.1%	79.1%	90%
mips	9,948	7,848	4.3%	30,094	(23%)	42.3%	54.5%	70.5%	90.9%
hardware	7,239	5,896	3.3%	21,878	(29%)	30.1%	42.1%	63.1%	88.2%
drivers/	5,970	5,269.6	2.9%	17,786	(32%)	30.3%	44.6%	59.7%	87.9%
arch/	684	132.7	8.1%	1,766	(6%)	13.4%	14.2%	79.6%	84.4%
sound/	585	493.6	5.3%	2,326	(18%)	41.3%	50.5%	76.6%	93.7%
software	2,709	1,952	7.4%	8,216	(7%)	74.8%	79.8%	90.3%	96.3%
kernel/	1,292	1,109.2	7.9%	4,292	(10%)	67.1%	74.4%	85.5%	94.6%
net/	1,029	662.3	6.8%	3,158	(1%)	86%	86.9%	98.7%	99.7%
rest	388	180.6	5.7%	766	(9%)	71.8%	78.3%	82.9%	90.3%
mn10300	8,746	7,308.2	3.6%	28,698	(33%)	39.5%	59.1%	60%	89.7%
hardware	6,091	5,423.4	2.7%	20,482	(40%)	26.6%	44.3%	51.7%	86.1%
drivers/	5,533	4,988.1	2.5%	17,786	(41%)	26.7%	45.1%	50.9%	86%
arch/	97	17.3	7.9%	370	(22%)	27.6%	34.5%	46.8%	58.3%
sound/	461	418	5.1%	2,326	(37%)	25.8%	40.7%	58.4%	92.4%
software	2,655	1,884.9	6.2%	8,216	(17%)	71.8%	85.6%	80.7%	96.1%
kernel/	1,246	1,051	6.4%	4,292	(24%)	61.8%	81%	71.8%	93.9%
net/	1,025	656.3	6.1%	3,158	(7%)	85.5%	91.6%	93.2%	99.7%
rest	384	177.6	5.5%	766	(13%)	71%	81.4%	79.5%	91%
openrisc	5,953	4,152.6	3.9%	28,367	(56%)	34%	77.6%	37.2%	84.9%
hardware	3,335	2,305.7	2.6%	20,151	(71%)	19%	65.2%	21.2%	72.6%
drivers/	3,059	2,075.1	2.3%	17,786	(71%)	18.4%	63.8%	20.6%	71.6%
arch/	25	6.3	1.7%	39	(8%)	71.8%	77.8%	71.8%	77.8%
sound/	251	224.2	5.2%	2,326	(69%)	23.1%	74.6%	24.7%	79.6%
software	2,618	1,846.9	5.6%	8,216	(21%)	70.7%	88.9%	76.3%	96%
kernel/	1,245	1,034.5	5.5%	4,292	(30%)	60.5%	85.6%	66.3%	93.8%
net/	992	636.4	6%	3,158	(10%)	84.6%	93.7%	90%	99.6%
rest	381	176	5.2%	766	(15%)	70.2%	82.7%	76.4%	90%
parisc	8,888	7,400.9	3.4%	28,748	(34%)	53.5%	80.7%	59.7%	90.1%
hardware	6,232	5,505.4	2.4%	20,532	(42%)	45.6%	78.3%	50.4%	86.6%
drivers/	5,684	5,050.4	2.2%	17,786	(42%)	45.9%	79.2%	50.3%	86.9%
arch/	86	39.6	9.7%	420	(1%)	39.3%	39.6%	60.7%	61.4%
sound/	462	415.4	3.9%	2,326	(47%)	44.4%	84%	49.1%	92.9%
software	2,656	1,895.6	6.4%	8,216	(14%)	73.3%	84.6%	83.1%	96%
kernel/	1,249	1,056.4	6.3%	4,292	(23%)	63.2%	81.2%	72.8%	93.7%
net/	1,021	660.4	6.8%	3,158	(2%)	87.1%	88.9%	97.7%	99.7%
rest	386	178.8	5.5%	766	(11%)	73%	81.9%	80.7%	90.4%
powerpc	10,089	8,055.1	4.7%	31,382	(20%)	58.5%	72.9%	71.9%	89.7%
hardware	7,369	6,089.9	3.8%	23,166	(25%)	49.8%	66.5%	64.9%	86.6%
drivers/	6,013	5,338	3%	17,786	(31%)	51.8%	74.3%	61.9%	88.9%
arch/	754	249.4	17.6%	3,054	(1%)	33.7%	34%	72%	72.7%
sound/	602	502.6	5.3%	2,326	(17%)	56.3%	67.5%	77.9%	93.5%
software	2,720	1,965.2	7.5%	8,216	(5%)	82.9%	87%	91.9%	96.5%
kernel/	1,303	1,122.1	8.2%	4,292	(8%)	76.8%	82.6%	88.2%	94.8%
net/	1,029	662.5	6.9%	3,158	(1%)	92.8%	93.7%	98.8%	99.7%
rest	388	180.6	5.7%	766	(8%)	75.6%	82.2%	84.1%	91.3%

Architecture	#files	Total kLOC	in #ifdef blocks	# variation points (dead/undead rate)	allyes CC _S	allyes CC _N	VAMPYR CC _S	VAMPYR CC _N
s390	3,946	2,782.9	5.7%	28,756 (67%)	24.2%	72.1%	26.8%	80%
hardware	1,526	1,034.5	2.8%	20,540 (86%)	5.1%	37.2%	5.8%	42.3%
drivers/	1,313	914.5	2.5%	17,786 (87%)	4.5%	35.4%	4.9%	38.9%
arch/	125	44.5	10.6%	428 (0%)	58.4%	58.7%	74.3%	74.4%
sound/	88	75.5	1.5%	2,326 (94%)	0%	0%	0%	0%
software	2,420	1,748.4	7.3%	8,216 (18%)	71.8%	86.8%	79.2%	95.8%
kernel/	1,216	1,054.6	7.6%	4,292 (17%)	69.7%	83.3%	78.7%	94.1%
net/	827	518.9	7.5%	3,158 (20%)	74.8%	92.9%	80%	99.4%
rest	377	174.9	5.6%	766 (13%)	71.5%	82%	79.2%	90.7%
score	5,784	4,035.8	4.1%	28,364 (58%)	32.9%	77.2%	36.2%	84.8%
hardware	3,150	2,164.9	2.4%	20,148 (73%)	16.9%	63.5%	18.9%	71%
drivers/	2,869	1,936.5	2.1%	17,786 (74%)	16.1%	61.8%	18.2%	69.7%
arch/	30	4.7	1.7%	36 (0%)	69.4%	69.4%	69.4%	69.4%
sound/	251	223.7	5%	2,326 (70%)	22.7%	74.7%	24.1%	79.2%
software	2,634	1,870.9	6%	8,216 (19%)	72.2%	88.3%	78.4%	96%
kernel/	1,259	1,057.5	6.1%	4,292 (26%)	63.5%	84.8%	70.2%	93.9%
net/	994	636.7	6%	3,158 (10%)	84.3%	93.5%	89.8%	99.6%
rest	381	176.6	5.4%	766 (14%)	71%	82.2%	77.5%	89.8%
sh	9,266	7,514.7	4.1%	29,166 (28%)	40.1%	55.8%	49.4%	68.8%
hardware	6,549	5,556.4	3%	20,950 (36%)	26.8%	41.9%	34.4%	53.7%
drivers/	5,690	5,036.2	2.7%	17,786 (38%)	27.5%	44.1%	33.9%	54.5%
arch/	381	92.7	5.7%	838 (1%)	15%	15.2%	56.1%	56.7%
sound/	478	427.5	5.2%	2,326 (35%)	25.9%	39.9%	30.3%	46.7%
software	2,717	1,958.2	7.2%	8,216 (9%)	73.8%	80.8%	87.7%	95.8%
kernel/	1,302	1,121.4	8%	4,292 (11%)	65.3%	72.9%	85.2%	94.7%
net/	1,029	657.3	6.1%	3,158 (6%)	85.9%	91.4%	92.5%	98.4%
rest	386	179.5	5.7%	766 (10%)	71.5%	79%	82.4%	90.9%
sparc	9,103	7,468.7	3.8%	28,987 (31%)	56.1%	81.5%	62%	90.1%
hardware	6,387	5,509.7	2.6%	20,771 (41%)	46%	78.2%	50.8%	86.4%
drivers/	5,663	5,001.1	2.3%	17,786 (42%)	46.6%	80.2%	50.5%	86.9%
arch/	261	89.9	8.6%	659 (1%)	34.3%	34.5%	65.4%	65.8%
sound/	463	418.7	4%	2,326 (47%)	44.7%	84.9%	49.1%	93.2%
software	2,716	1,959	7.4%	8,216 (6%)	81.7%	86.7%	90.4%	95.9%
kernel/	1,300	1,116.3	8%	4,292 (9%)	75.1%	82%	85.9%	93.8%
net/	1,028	662.1	6.8%	3,158 (1%)	92.6%	93.8%	98.4%	99.7%
rest	388	180.6	5.7%	766 (9%)	74.3%	81.3%	82.5%	90.2%
unicore32	8,726	7,261.4	3.6%	28,460 (34%)	55.2%	82.9%	60.2%	90.4%
hardware	6,077	5,395	2.9%	20,244 (39%)	48.4%	79.4%	53%	87.1%
drivers/	5,557	4,965.3	2.7%	17,786 (40%)	47.7%	79.2%	52.3%	86.8%
arch/	59	11	8.9%	132 (2%)	43.2%	44.2%	49.2%	50.4%
sound/	461	418.8	5.3%	2,326 (36%)	53.3%	83.7%	58.9%	92.5%
software	2,649	1,866.4	5.7%	8,216 (20%)	72.1%	89.3%	77.9%	96.5%
kernel/	1,245	1,034.9	5.6%	4,292 (30%)	61.4%	86.3%	67.3%	94.6%
net/	1,023	654.8	5.9%	3,158 (8%)	86.6%	93.7%	92.1%	99.7%
rest	381	176.7	5.4%	766 (14%)	71.7%	83.3%	77.9%	90.6%
x86	10,376	8,390.6	4.5%	30,368 (17%)	65.2%	78.6%	73.3%	88.4%
hardware	7,643	6,416.8	3.6%	22,152 (22%)	59.7%	76.8%	67.3%	86.5%
drivers/	6,525	5,721.6	3.1%	17,786 (26%)	60%	80.4%	67%	89.8%
arch/	528	191.1	13%	2,040 (0%)	45.9%	46%	59.2%	59.3%
sound/	590	504.1	5.3%	2,326 (18%)	69.1%	84.2%	76.3%	92.9%
software	2,733	1,973.8	7.6%	8,216 (4%)	80.2%	82.7%	89.6%	92.4%
kernel/	1,316	1,130.6	8.3%	4,292 (5%)	75.3%	78.5%	87.5%	91.4%

Architecture	#files	Total kLOC	in #ifdef blocks	# variation points (dead/undead rate)	allyes CC_S	allyes CC_N	VAMPYR CC_S	VAMPYR CC_N	
net/	1,028	662.2	6.9%	3,158	(1%)	87.9%	88.7%	93.7%	94.6%
rest	389	181	5.8%	766	(5%)	75.7%	80%	84.3%	89.1%
xtensa	8,762	7,238.5	3.2%	28,429	(38%)	51.8%	82.8%	56.4%	90.3%
hardware	6,115	5,374.9	2.4%	20,213	(45%)	43.9%	79.4%	48.1%	87%
drivers/	5,602	4,949.8	2.2%	17,786	(45%)	44%	79.1%	48.1%	86.5%
arch/	50	10.5	5.1%	101	(10%)	46.5%	51.6%	51.5%	57.1%
sound/	463	414.6	4%	2,326	(48%)	43.5%	84.2%	48%	93.1%
software	2,647	1,863.5	5.5%	8,216	(20%)	71%	88.5%	77%	96%
kernel/	1,243	1,032.5	5.4%	4,292	(30%)	59.8%	84.9%	66.1%	93.8%
net/	1,022	654.4	5.9%	3,158	(8%)	86.4%	93.7%	91.8%	99.7%
rest	382	176.6	5.2%	766	(14%)	70.5%	82.1%	76.9%	89.6%

Table 1: Comprehensive coverage data for Linux v3.2: rate (defect corrected) of covered variation points for allyesconfig and for the VAMPYR approach.