2020 공개SW 페스티벌

Linux Kernel and Contribution

LG Electronics
박병철

Who I am

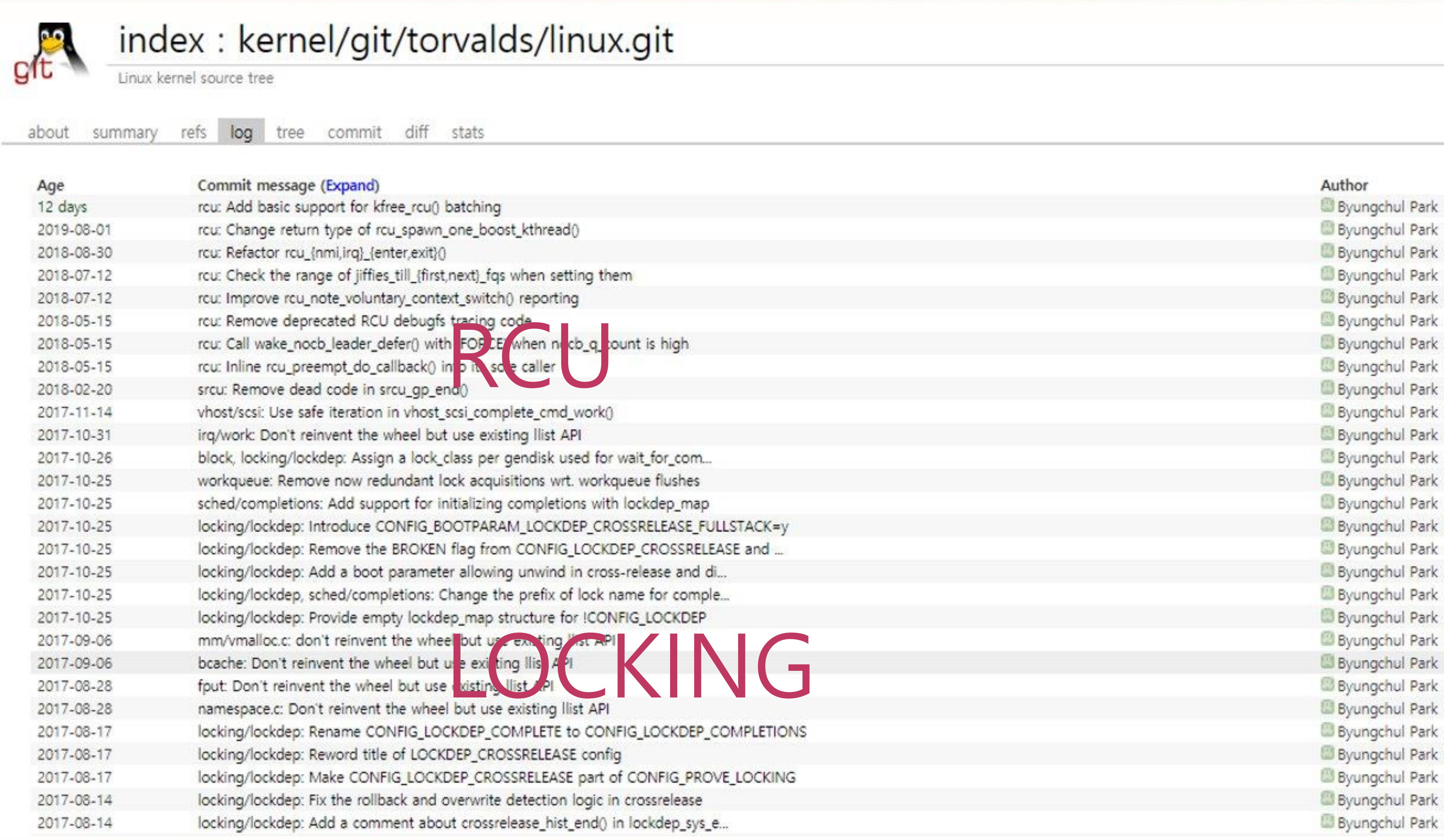
Byungchul Park

Open Source Contribution Part

CTO division, LG Electronics

Linux kernel 17 years

Linux kernel contribution 7 years



index : kernel/git/torvalds/linux.git

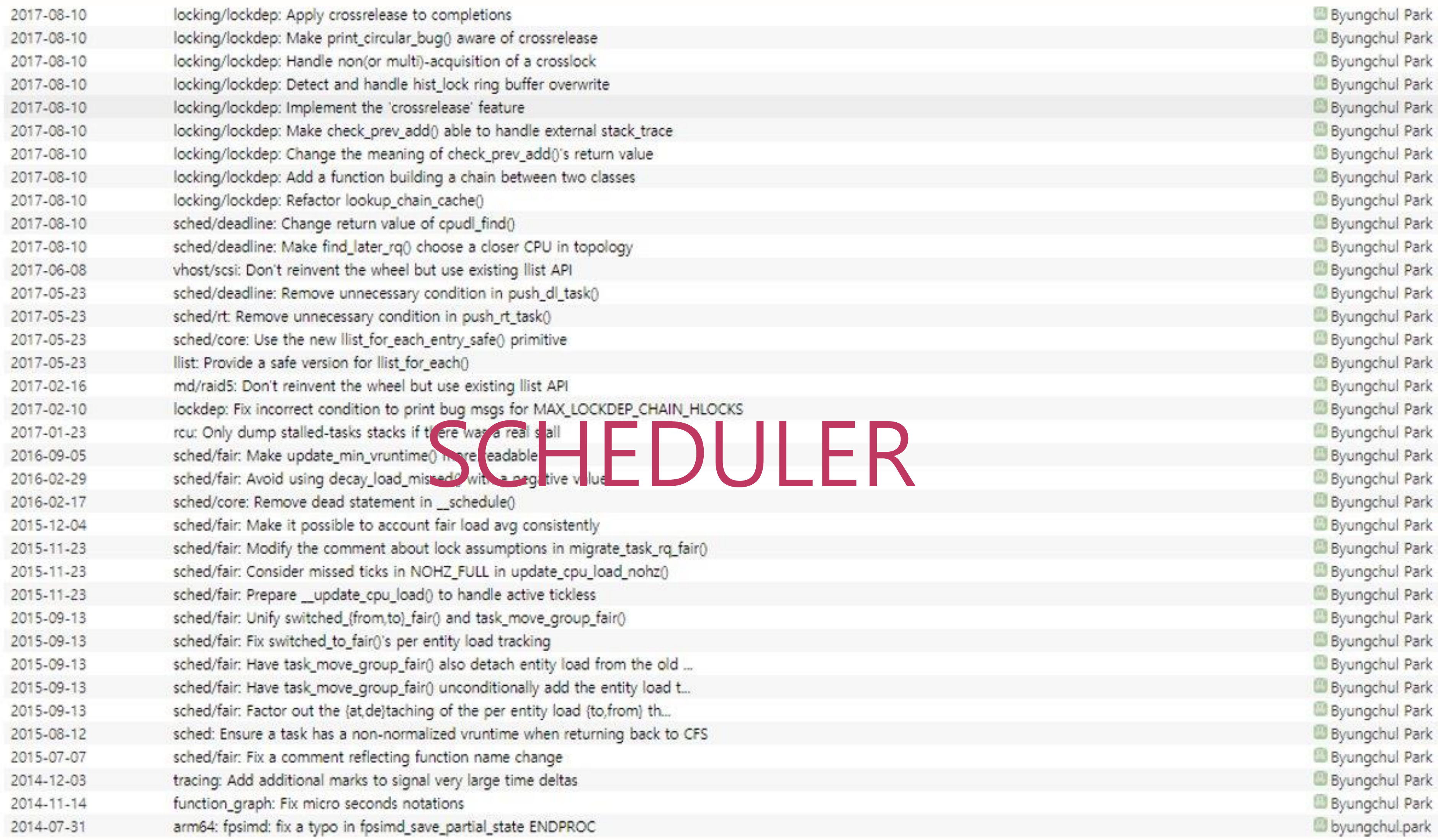
Linux kernel source tree

about summary refs log tree commit diff stats

Age	Commit message (Expand)	Author
12 days	rcu: Add basic support for kfree_rcu() batching	Byungchul Park
2019-08-01	rcu: Change return type of rcu_spawn_one_boost_kthread()	Byungchul Park
2018-08-30	rcu: Refactor rcu_{nmi,irq}_{enter,exit}()	Byungchul Park
2018-07-12	rcu: Check the range of jiffies_till_{first,next}_fqs when setting them	Byungchul Park
2018-07-12	rcu: Improve rcu_note_voluntary_context_switch() reporting	Byungchul Park
2018-05-15	rcu: Remove deprecated RCU debugfs tracing code	Byungchul Park
2018-05-15	rcu: Call wake_nocb_leader_defer() with FORCE when nocb_q_count is high	Byungchul Park
2018-05-15	rcu: Inline rcu_preempt_do_callback() into its sole caller	Byungchul Park
2018-02-20	srcu: Remove dead code in srcu_gp_end()	Byungchul Park
2017-11-14	vhost/scsi: Use safe iteration in vhost_scsi_complete_cmd_work()	Byungchul Park
2017-10-31	irq/work: Don't reinvent the wheel but use existing llist API	Byungchul Park
2017-10-26	block, locking/lockdep: Assign a lock_class per gendisk used for wait_for_com...	Byungchul Park
2017-10-25	workqueue: Remove now redundant lock acquisitions wrt. workqueue flushes	Byungchul Park
2017-10-25	sched/completions: Add support for initializing completions with lockdep_map	Byungchul Park
2017-10-25	locking/lockdep: Introduce CONFIG_BOOTPARAM_LOCKDEP_CROSSRELEASE_FULLSTACK=y	Byungchul Park
2017-10-25	locking/lockdep: Remove the BROKEN flag from CONFIG_LOCKDEP_CROSSRELEASE and ...	Byungchul Park
2017-10-25	locking/lockdep: Add a boot parameter allowing unwind in cross-release and di...	Byungchul Park
2017-10-25	locking/lockdep, sched/completions: Change the prefix of lock name for comple...	Byungchul Park
2017-10-25	locking/lockdep: Provide empty lockdep_map structure for !CONFIG_LOCKDEP	Byungchul Park
2017-09-06	mm/vmalloc.c: don't reinvent the wheel but use existing llist API	Byungchul Park
2017-09-06	bcache: Don't reinvent the wheel but use existing llist API	Byungchul Park
2017-08-28	fput: Don't reinvent the wheel but use existing llist API	Byungchul Park
2017-08-28	namespace.c: Don't reinvent the wheel but use existing llist API	Byungchul Park
2017-08-17	locking/lockdep: Rename CONFIG_LOCKDEP_COMPLETE to CONFIG_LOCKDEP_COMPLETIONS	Byungchul Park
2017-08-17	locking/lockdep: Reword title of LOCKDEP_CROSSRELEASE config	Byungchul Park
2017-08-17	locking/lockdep: Make CONFIG_LOCKDEP_CROSSRELEASE part of CONFIG_PROVE_LOCKING	Byungchul Park
2017-08-14	locking/lockdep: Fix the rollback and overwrite detection logic in crossrelease	Byungchul Park
2017-08-14	locking/lockdep: Add a comment about crossrelease_hist_end() in lockdep_sys_e...	Byungchul Park

RCU

LOCKING



2017-08-10	locking/lockdep: Apply crossrelease to completions	Byungchul Park
2017-08-10	locking/lockdep: Make print_circular_bug() aware of crossrelease	Byungchul Park
2017-08-10	locking/lockdep: Handle non(or multi)-acquisition of a crosslock	Byungchul Park
2017-08-10	locking/lockdep: Detect and handle hist_lock ring buffer overwrite	Byungchul Park
2017-08-10	locking/lockdep: Implement the 'crossrelease' feature	Byungchul Park
2017-08-10	locking/lockdep: Make check_prev_add() able to handle external stack_trace	Byungchul Park
2017-08-10	locking/lockdep: Change the meaning of check_prev_add()'s return value	Byungchul Park
2017-08-10	locking/lockdep: Add a function building a chain between two classes	Byungchul Park
2017-08-10	locking/lockdep: Refactor lookup_chain_cache()	Byungchul Park
2017-08-10	sched/deadline: Change return value of cpudl_find()	Byungchul Park
2017-08-10	sched/deadline: Make find_later_rq() choose a closer CPU in topology	Byungchul Park
2017-06-08	vhost/scsi: Don't reinvent the wheel but use existing llist API	Byungchul Park
2017-05-23	sched/deadline: Remove unnecessary condition in push_dl_task()	Byungchul Park
2017-05-23	sched/rt: Remove unnecessary condition in push_rt_task()	Byungchul Park
2017-05-23	sched/core: Use the new llist_for_each_entry_safe() primitive	Byungchul Park
2017-05-23	llist: Provide a safe version for llist_for_each()	Byungchul Park
2017-02-16	md/raid5: Don't reinvent the wheel but use existing llist API	Byungchul Park
2017-02-10	lockdep: Fix incorrect condition to print bug msgs for MAX_LOCKDEP_CHAIN_HLOCKS	Byungchul Park
2017-01-23	rcu: Only dump stalled-tasks stacks if there was a real stall	Byungchul Park
2016-09-05	sched/fair: Make update_min_vruntime() more readable	Byungchul Park
2016-02-29	sched/fair: Avoid using decay_load_missed() with a negative value	Byungchul Park
2016-02-17	sched/core: Remove dead statement in __schedule()	Byungchul Park
2015-12-04	sched/fair: Make it possible to account fair load avg consistently	Byungchul Park
2015-11-23	sched/fair: Modify the comment about lock assumptions in migrate_task_rq_fair()	Byungchul Park
2015-11-23	sched/fair: Consider missed ticks in NOHZ_FULL in update_cpu_load_nohz()	Byungchul Park
2015-11-23	sched/fair: Prepare __update_cpu_load() to handle active tickless	Byungchul Park
2015-09-13	sched/fair: Unify switched_{from,to}_fair() and task_move_group_fair()	Byungchul Park
2015-09-13	sched/fair: Fix switched_to_fair()'s per entity load tracking	Byungchul Park
2015-09-13	sched/fair: Have task_move_group_fair() also detach entity load from the old ...	Byungchul Park
2015-09-13	sched/fair: Have task_move_group_fair() unconditionally add the entity load t...	Byungchul Park
2015-09-13	sched/fair: Factor out the {at,de}taching of the per entity load {to,from} th...	Byungchul Park
2015-08-12	sched: Ensure a task has a non-normalized vruntime when returning back to CFS	Byungchul Park
2015-07-07	sched/fair: Fix a comment reflecting function name change	Byungchul Park
2014-12-03	tracing: Add additional marks to signal very large time deltas	Byungchul Park
2014-11-14	function_graph: Fix micro seconds notations	Byungchul Park
2014-07-31	arm64: fpsimd: fix a typo in fpsimd_save_partial_state ENDPROC	byungchul.park

SCHEDULER

BRONCOS

Ke ne

BRONCOS

Ke ne

BRONCOS

BRONCOS

Ke ne

BRONCOS

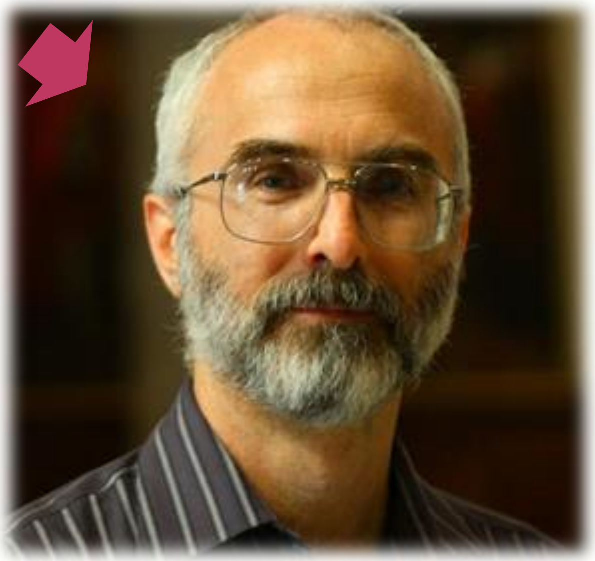
Kerne

BRONCOS

Ke ne

BRONCOS

Ke ne



OSPM SUMMIT

OSPM SUMMIT

KERNEL SUMMIT

KERNEL SUMMIT

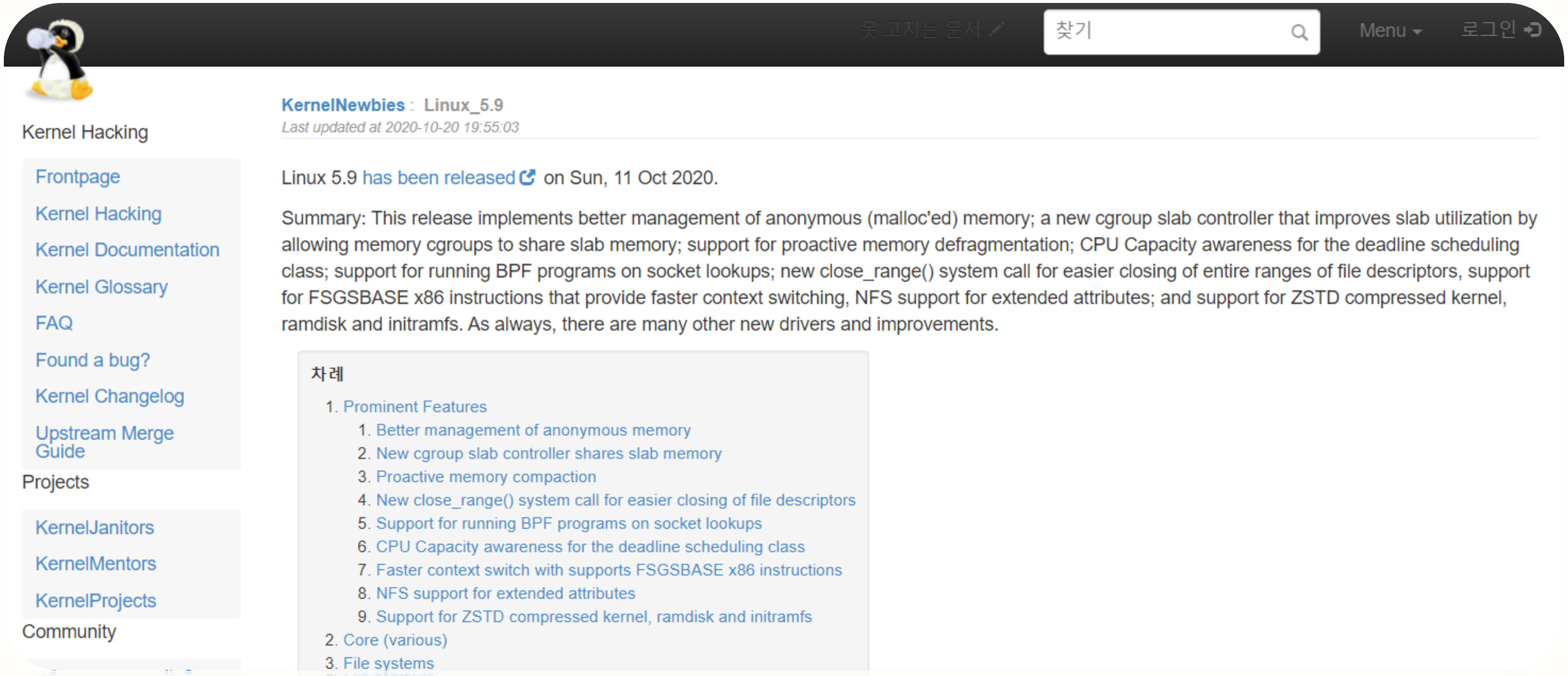
Study

오픈소스로 여는 뉴노멀
2020
공개SW
페스티벌

max.byungchul.park@gmail.com

Subsystems

Core	task scheduler, locking, rcu, workqueue
Memory	allocator, reclaim, swap, zram, oom
Filesystem	vfs, ext4, f2fs
Block	io scheduler
Architecture	arm, arm64, x86, x64
Trace	ftrace, ebpf
Driver	staging, device drivers
Network	internet
Security	selinux, smack, yama



Kernel Hacking

- Frontpage
- Kernel Hacking
- Kernel Documentation
- Kernel Glossary
- FAQ
- Found a bug?
- Kernel Changelog
- Upstream Merge Guide

Projects

- KernelJanitors
- KernelMentors
- KernelProjects

Community

KernelNewbies : Linux_5.9

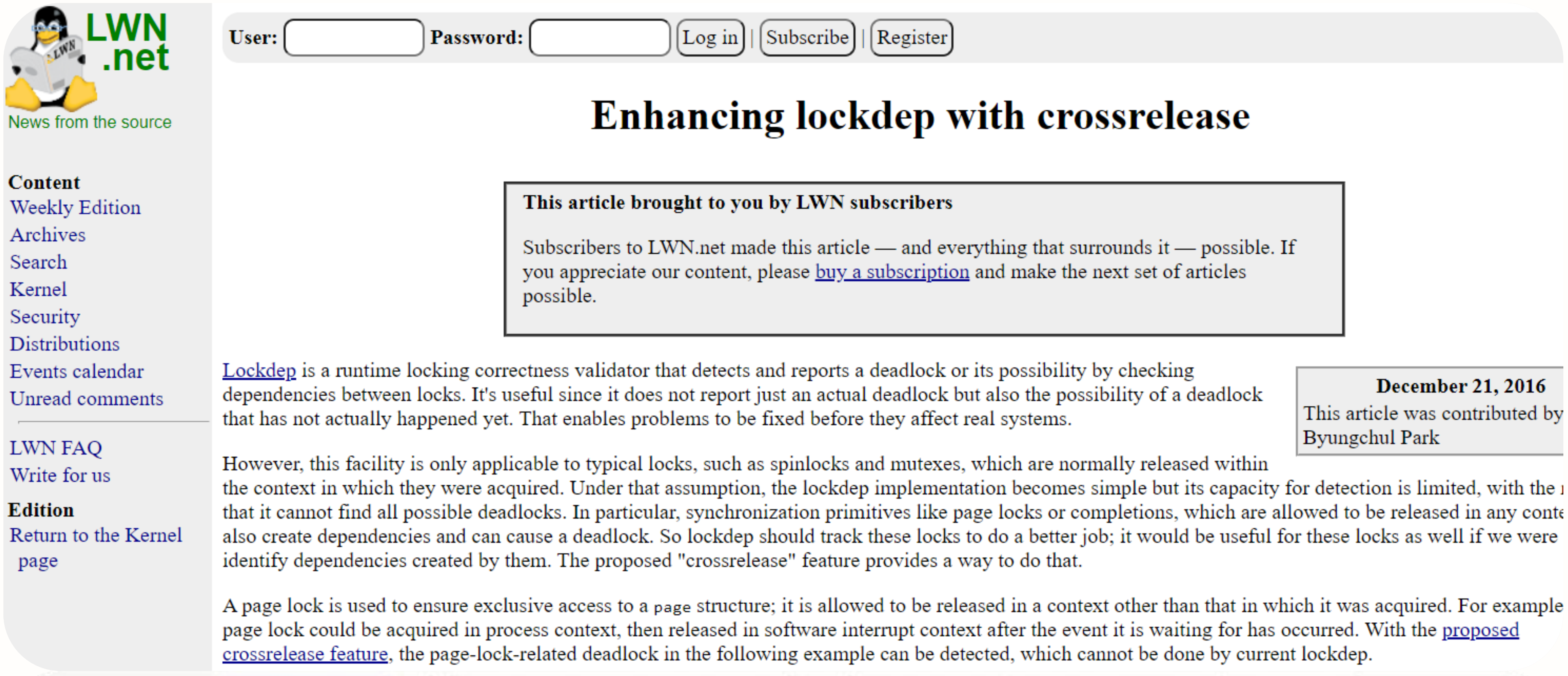
Last updated at 2020-10-20 19:55:03

Linux 5.9 has been released on Sun, 11 Oct 2020.

Summary: This release implements better management of anonymous (malloc'ed) memory; a new cgroup slab controller that improves slab utilization by allowing memory cgroups to share slab memory; support for proactive memory defragmentation; CPU Capacity awareness for the deadline scheduling class; support for running BPF programs on socket lookups; new close_range() system call for easier closing of entire ranges of file descriptors, support for FSGSBASE x86 instructions that provide faster context switching, NFS support for extended attributes; and support for ZSTD compressed kernel, ramdisk and initramfs. As always, there are many other new drivers and improvements.

차례

kernelnewbies.org



News from the source

Content
Weekly Edition
Archives
Search
Kernel
Security
Distributions
Events calendar
Unread comments

LWN FAQ
Write for us

Edition
Return to the Kernel page

User: Password: Log in | Subscribe | Register

Enhancing lockdep with crossrelease

This article brought to you by LWN subscribers

Subscribers to LWN.net made this article — and everything that surrounds it — possible. If you appreciate our content, please buy a subscription and make the next set of articles possible.

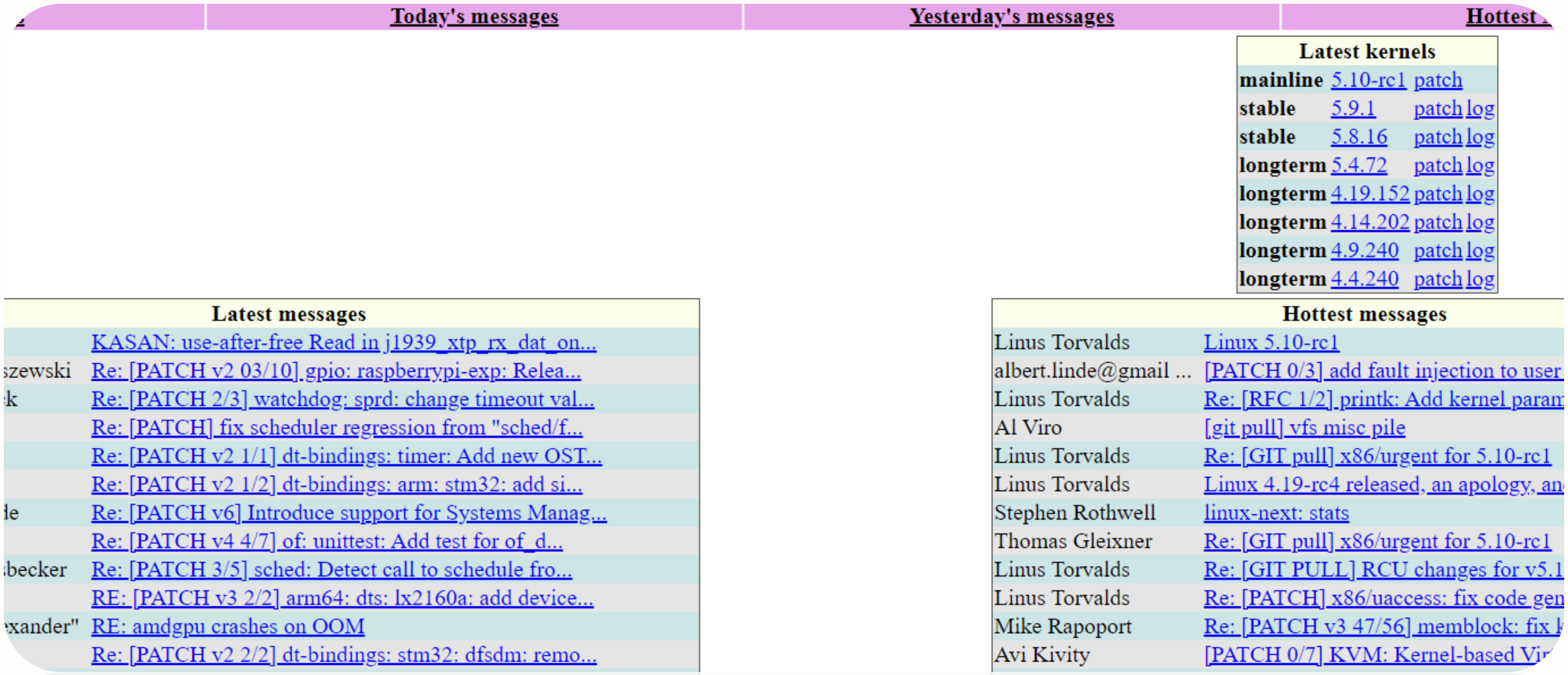
December 21, 2016
This article was contributed by Byungchul Park

Lockdep is a runtime locking correctness validator that detects and reports a deadlock or its possibility by checking dependencies between locks. It's useful since it does not report just an actual deadlock but also the possibility of a deadlock that has not actually happened yet. That enables problems to be fixed before they affect real systems.

However, this facility is only applicable to typical locks, such as spinlocks and mutexes, which are normally released within the context in which they were acquired. Under that assumption, the lockdep implementation becomes simple but its capacity for detection is limited, with the that it cannot find all possible deadlocks. In particular, synchronization primitives like page locks or completions, which are allowed to be released in any conte also create dependencies and can cause a deadlock. So lockdep should track these locks to do a better job; it would be useful for these locks as well if we were identify dependencies created by them. The proposed "crossrelease" feature provides a way to do that.

A page lock is used to ensure exclusive access to a `page` structure; it is allowed to be released in a context other than that in which it was acquired. For example page lock could be acquired in process context, then released in software interrupt context after the event it is waiting for has occurred. With the proposed crossrelease feature, the page-lock-related deadlock in the following example can be detected, which cannot be done by current lockdep.

lwn.net



	Today's messages	Yesterday's messages	Hottest

Latest kernels			
mainline	5.10-rc1	patch	
stable	5.9.1	patch	log
stable	5.8.16	patch	log
longterm	5.4.72	patch	log
longterm	4.19.152	patch	log
longterm	4.14.202	patch	log
longterm	4.9.240	patch	log
longterm	4.4.240	patch	log

Latest messages	
	KASAN: use-after-free Read in j1939_xtp_rx_dat_on...
szewski	Re: [PATCH v2 03/10] gpio: raspberrypi-exp: Relea...
k	Re: [PATCH 2/3] watchdog: sprd: change timeout val...
	Re: [PATCH] fix scheduler regression from "sched/f...
	Re: [PATCH v2 1/1] dt-bindings: timer: Add new OST...
	Re: [PATCH v2 1/2] dt-bindings: arm: stm32: add si...
le	Re: [PATCH v6] Introduce support for Systems Manag...
	Re: [PATCH v4 4/7] of: unittest: Add test for of_d...
becker	Re: [PATCH 3/5] sched: Detect call to schedule fro...
	RE: [PATCH v3 2/2] arm64: dts: lx2160a: add device...
xander"	RE: amdgpu crashes on OOM
	Re: [PATCH v2 2/2] dt-bindings: stm32: dfsdm: remo...

Hottest messages	
Linus Torvalds	Linux 5.10-rc1
albert.linde@gmail ...	[PATCH 0/3] add fault injection to user
Linus Torvalds	Re: [RFC 1/2] printk: Add kernel param
Al Viro	[git pull] vfs misc pile
Linus Torvalds	Re: [GIT pull] x86/urgent for 5.10-rc1
Linus Torvalds	Linux 4.19-rc4 released, an apology, an
Stephen Rothwell	linux-next: stats
Thomas Gleixner	Re: [GIT pull] x86/urgent for 5.10-rc1
Linus Torvalds	Re: [GIT PULL] RCU changes for v5.1
Linus Torvalds	Re: [PATCH] x86/uaccess: fix code gen
Mike Rapoport	Re: [PATCH v3 47/56] memblock: fix k
Avi Kivity	[PATCH 0/7] KVM: Kernel-based Vir

lkml.org

Contribution

How it works

Linus Torvalds

Maintainers / Reviewers

Contributors

How it works

Linus Torvalds

Maintainers / Reviewers

Contributors

How it works

Linus Torvalds

Maintainers / Reviewers

Contributors

Subsystems

Core	task scheduler, locking, rcu, workqueue
Memory	allocator, reclaim, swap, zram, oom
Filesystem	vfs, ext4, f2fs
Block	io scheduler
Architecture	arm, arm64, x86, x64
Trace	ftrace, ebpf
Driver	staging, device drivers
Network	internet
Security	selinux, smack, yama

BRONCOS

Ke ne

How it works

Linus Torvalds

Maintainers / Reviewers

Contributors

Setup

LKML subscription

Plain text email client

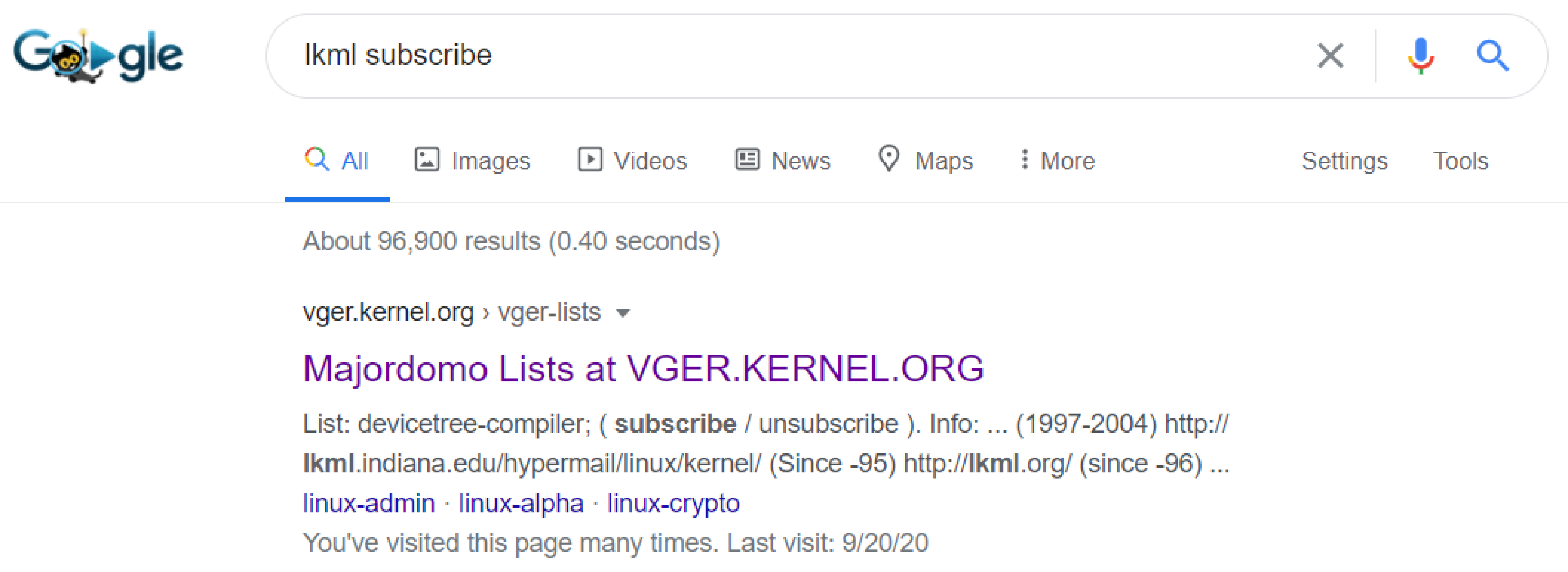
Git configuration

Setup

LKML subscription

Plain text email client

Git configuration



About 96,900 results (0.40 seconds)

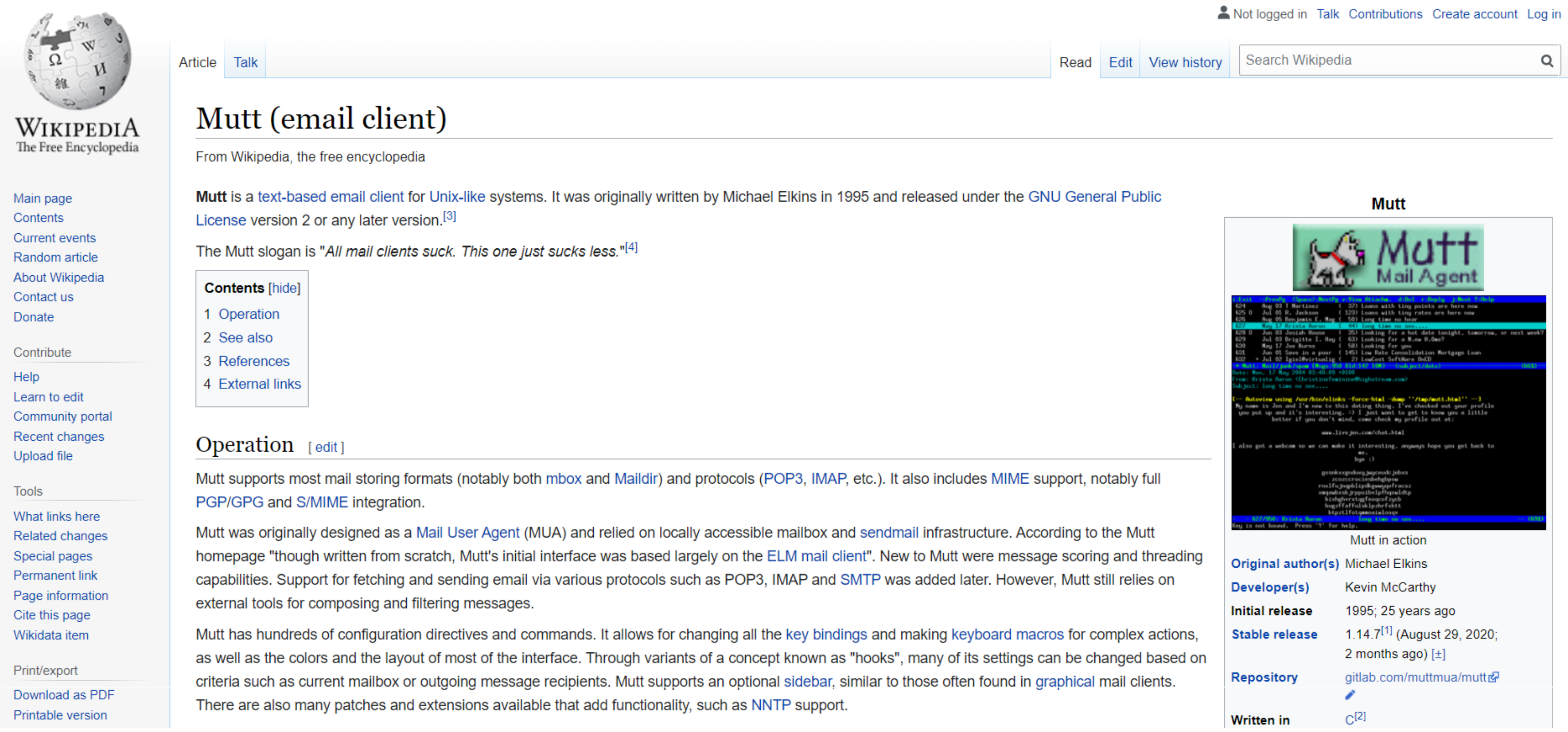
vger.kernel.org › vger-lists

Majordomo Lists at VGER.KERNEL.ORG

List: devicetree-compiler; (**subscribe** / unsubscribe). Info: ... (1997-2004) http://**lkml**.indiana.edu/hypermail/linux/kernel/ (Since -95) http://**lkml**.org/ (since -96) ...

linux-admin · linux-alpha · linux-crypto

You've visited this page many times. Last visit: 9/20/20



Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

Mutt (email client)

From Wikipedia, the free encyclopedia

Mutt is a text-based email client for Unix-like systems. It was originally written by Michael Elkins in 1995 and released under the GNU General Public License version 2 or any later version.[3]

The Mutt slogan is "*All mail clients suck. This one just sucks less.*"[4]

Contents [hide]

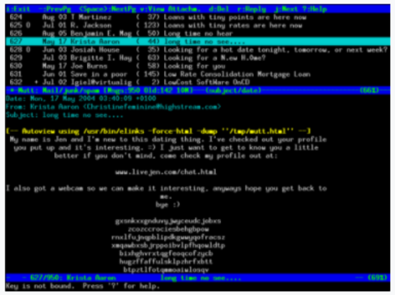
Operation [edit]

Mutt supports most mail storing formats (notably both mbox and Maildir) and protocols (POP3, IMAP, etc.). It also includes MIME support, notably full PGP/GPG and S/MIME integration.

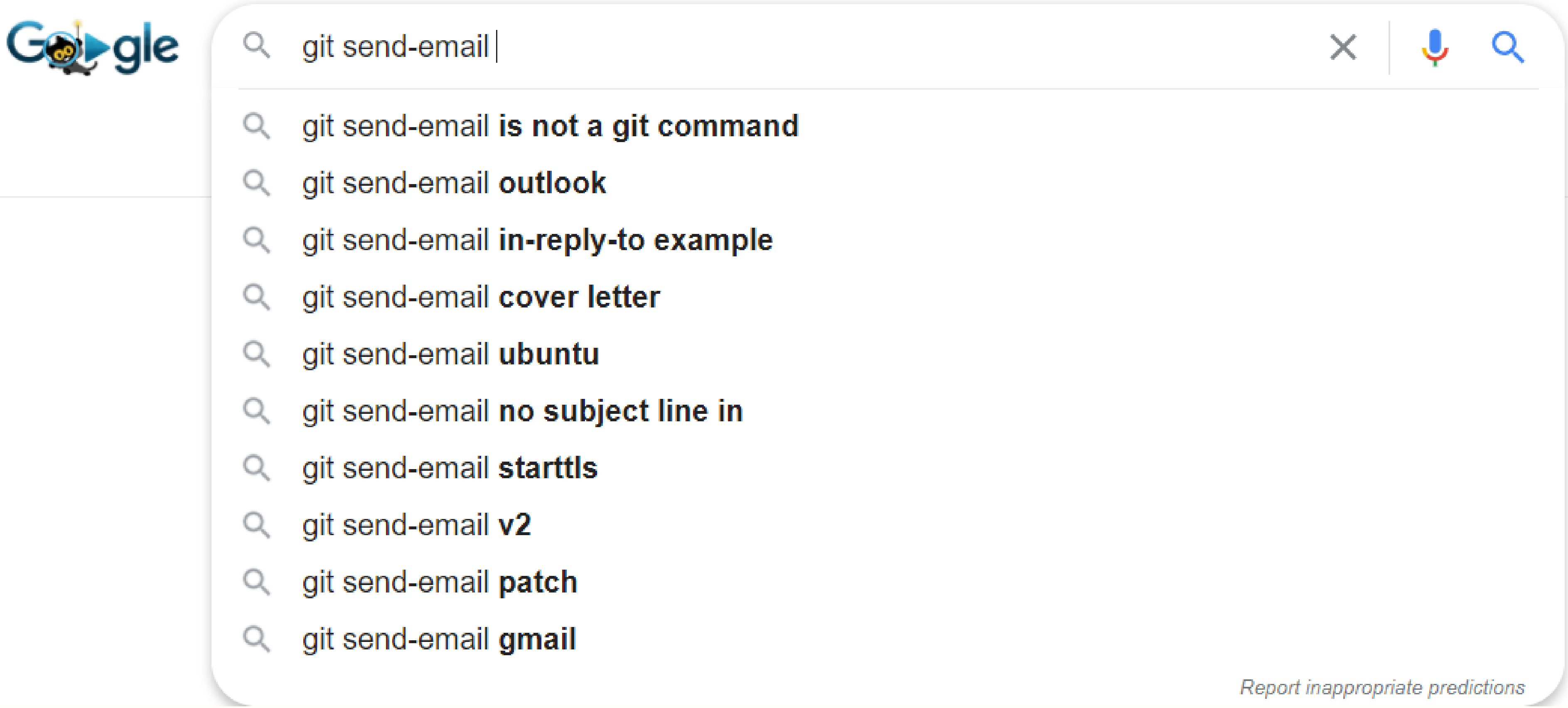
Mutt was originally designed as a Mail User Agent (MUA) and relied on locally accessible mailbox and sendmail infrastructure. According to the Mutt homepage "though written from scratch, Mutt's initial interface was based largely on the ELM mail client". New to Mutt were message scoring and threading capabilities. Support for fetching and sending email via various protocols such as POP3, IMAP and SMTP was added later. However, Mutt still relies on external tools for composing and filtering messages.

Mutt has hundreds of configuration directives and commands. It allows for changing all the key bindings and making keyboard macros for complex actions, as well as the colors and the layout of most of the interface. Through variants of a concept known as "hooks", many of its settings can be changed based on criteria such as current mailbox or outgoing message recipients. Mutt supports an optional sidebar, similar to those often found in graphical mail clients. There are also many patches and extensions available that add functionality, such as NNTP support.

Mutt


Mutt in action

Original author(s)	Michael Elkins
Developer(s)	Kevin McCarthy
Initial release	1995; 25 years ago
Stable release	1.14.7[1] (August 29, 2020; 2 months ago) [±]
Repository	gitlab.com/muttmua/mutt
Written in	C[2]



git send-email

git send-email **is not a git command**
git send-email **outlook**
git send-email **in-reply-to example**
git send-email **cover letter**
git send-email **ubuntu**
git send-email **no subject line in**
git send-email **starttls**
git send-email **v2**
git send-email **patch**
git send-email **gmail**

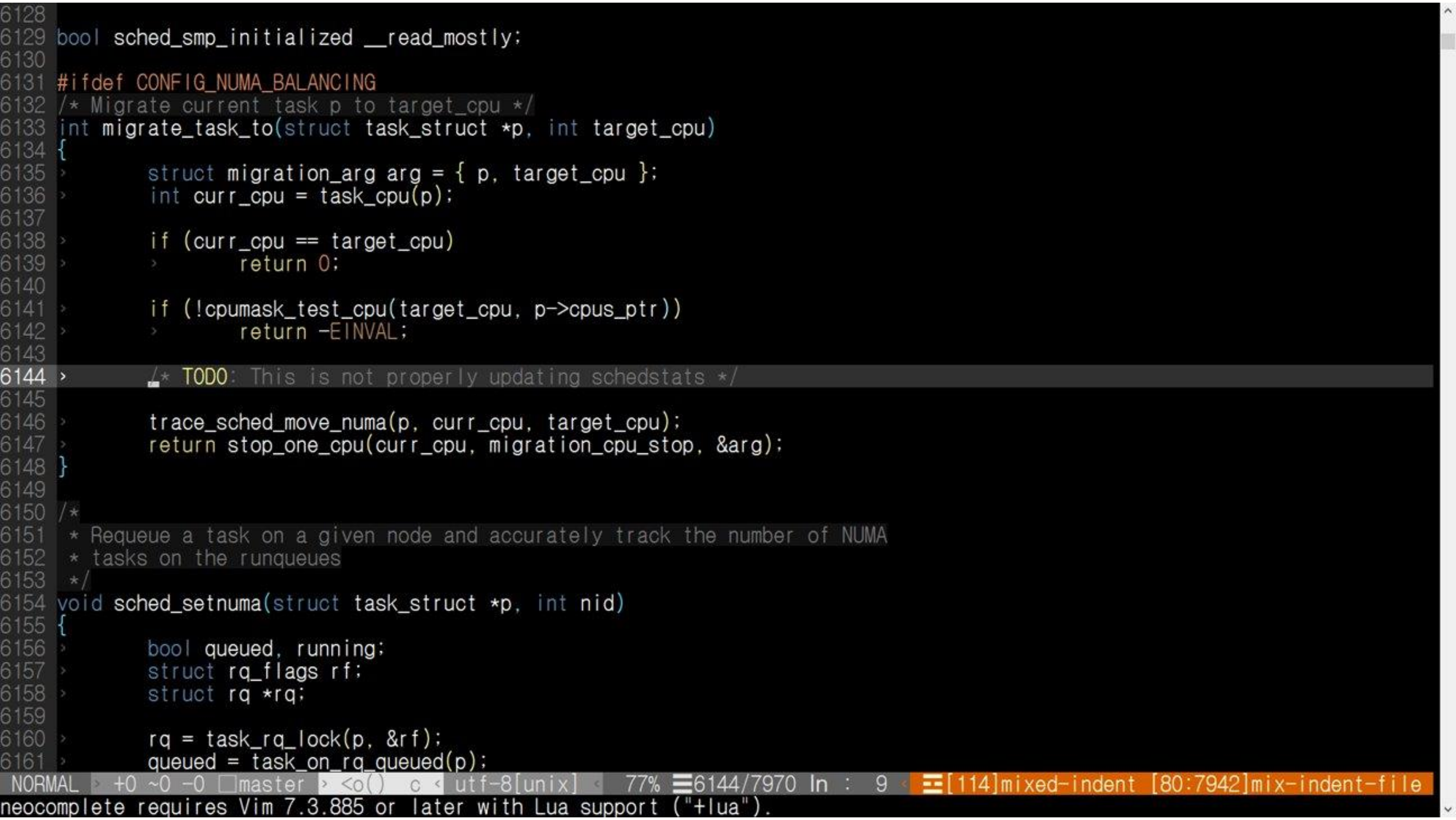
Report inappropriate predictions

Items

Typo

FIXME / TODO / XXX

Read the code



```
bool sched_smp_initialized __read_mostly;

#ifdef CONFIG_NUMA_BALANCING
/* Migrate current task p to target_cpu */
int migrate_task_to(struct task_struct *p, int target_cpu)
{
	struct migration_arg arg = { p, target_cpu };
	int curr_cpu = task_cpu(p);

	if (curr_cpu == target_cpu)
		return 0;

	if (!cpumask_test_cpu(target_cpu, p->cpus_ptr))
		return -EINVAL;

	/* TODO: This is not properly updating schedstats */

	trace_sched_move_numa(p, curr_cpu, target_cpu);
	return stop_one_cpu(curr_cpu, migration_cpu_stop, &arg);
}

/*
 * Requeue a task on a given node and accurately track the number of NUMA
 * tasks on the runqueues
 */
void sched_setnuma(struct task_struct *p, int nid)
{
	bool queued, running;
	struct rq_flags rf;
	struct rq *rq;

	rq = task_rq_lock(p, &rf);
	queued = task_on_rq_queued(p);
 NORMAL  +0 ~0 -0  master  <o()  c  utf-8[unix]  77% ☰6144/7970 ln : 9  ☲[114]mixed-indent [80:7942]mix-indent-file
neocomplete requires Vim 7.3.885 or later with Lua support ("+lua").
```

Items

Typo

FIXME / TODO / XXX

Read the code

Items

Typo

FIXME / TODO / XXX

Read the code

Must read

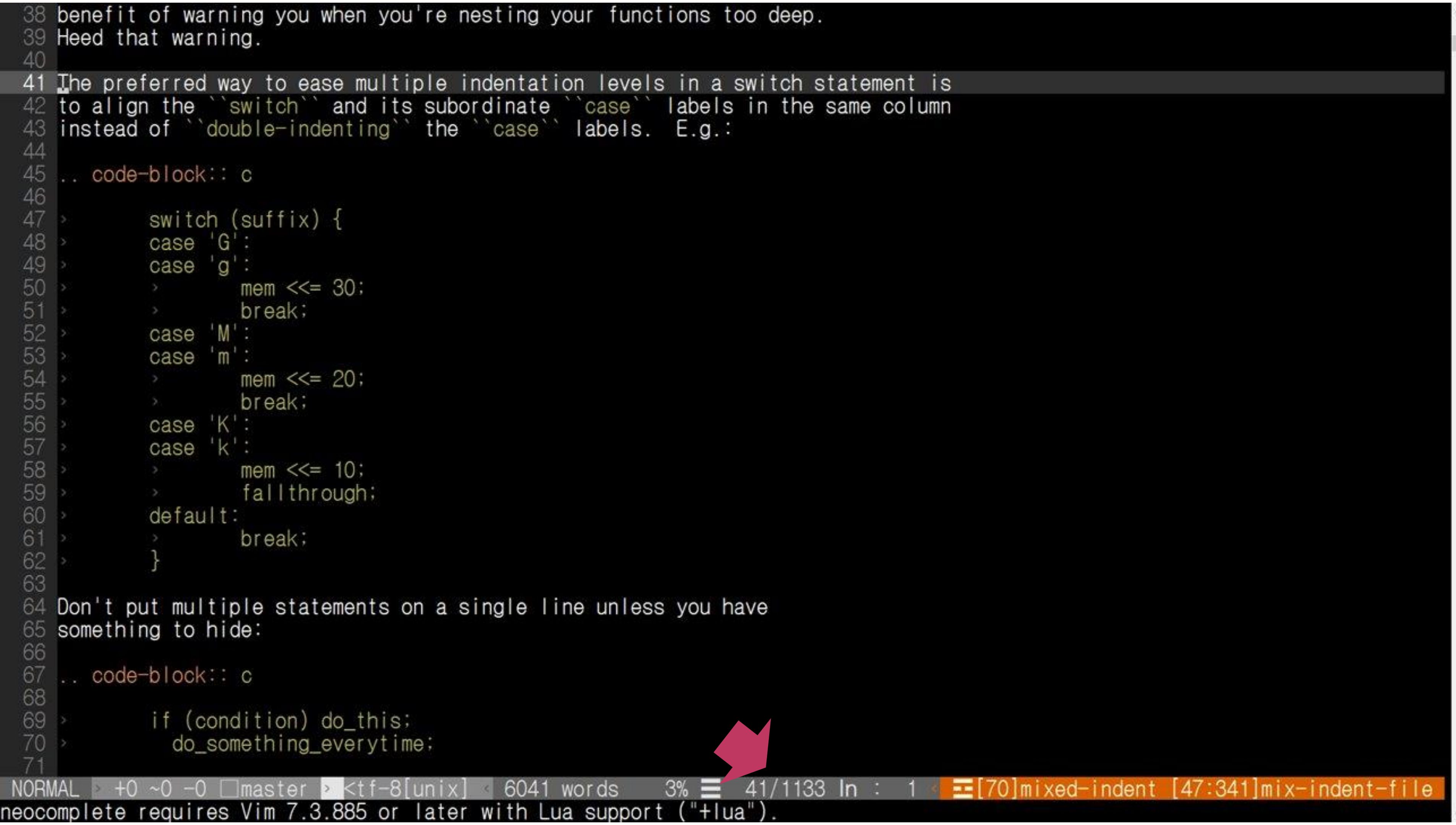
Documentation/process/*

Documentation/process/coding-style.rst

Must read

Documentation/process/*

Documentation/process/coding-style.rst



```
benefit of warning you when you're nesting your functions too deep.
Heed that warning.

The preferred way to ease multiple indentation levels in a switch statement is
to align the ``switch`` and its subordinate ``case`` labels in the same column
instead of ``double-indenting`` the ``case`` labels.  E.g.:

.. code-block:: c

        switch (suffix) {
        case 'G':
        case 'g':
                mem <<= 30;
                break;
        case 'M':
        case 'm':
                mem <<= 20;
                break;
        case 'K':
        case 'k':
                mem <<= 10;
                fallthrough;
        default:
                break;
        }

Don't put multiple statements on a single line unless you have
something to hide:

.. code-block:: c

        if (condition) do_this;
          do_something_everytime;

NORMAL  +0 ~0 -0  master  <tf-8[unix]  6041 words  3%  41/1133 ln :  1  [70]mixed-indent [47:341]mix-indent-file
neocomplete requires Vim 7.3.885 or later with Lua support ("+lua").
```

Communication

Use plain text email client

Inline posting

Get used to arrogant guys

Communication

Use plain text email client

Inline posting

Get used to arrogant guys

Subsystems

Core	task scheduler, locking, rcu, workqueue
Memory	allocator, reclaim, swap, zram, oom
Filesystem	vfs, ext4, f2fs
Block	io scheduler
Architecture	arm, arm64, x86, x64
Trace	ftrace, ebpf
Driver	staging, device drivers
Network	internet
Security	selinux, smack, yama

Thank you

오픈소스로 여는 뉴노멀

2020 공개SW 페스티벌

과학기술정보통신부
Ministry of Science and ICT

nipa 정보통신산업진흥원
National IT Industry Promotion Agency