LINUX系统高级调试和优化

--LINUX大局观

张银奎

2016/8/18(2017/8/14更新于松雪楼, 2018/4更新于863, 2019/3)

1

Linus Benedict Torvalds

Linus Benedict Torvalds (/ˈlaɪnəsˈtɔːrvɔːldz/;[5] Swedish: [ˈliːn.ɵs ˈtuːr.valds]; born December 28, 1969) is a Finnish software engineer, American naturalized, who is the creator and, for a long time, principal developer, of the Linux kernel, which became the kernel for operating systems (and many distributions of each) such as GNU and years later Android and Chrome OS. He also created the distributed revision control system git. He was honored, along with Shinya Yamanaka, with the 2012 Millennium Technology Prize by the Technology Academy Finland "in recognition of his creation of a new open source operating system for computers leading to the widely used Linux kernel". He is also the recipient of the 2014 IEEE Computer Society Computer Pioneer Award.

-- https://en.wikipedia.org/wiki/Linus_Torvalds

Torvalds at LinuxCon Europe 2014

2

From: torv...@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Keywords: 386, preferences
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
Lines: 20

Hello everybody out there using minix -
I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).
I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)
Linus (torvalds@kruuna.helsinki.fi)
PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.
— Linus Torvalds

3

版本历史

Version	Release Date	Total Size	Size of mm	Line Count
0.01	1991/8/25			
0.11	1991/12/8	319681B	12544B	
1.0	March 13, 1992	5.9MiB	96KiB	3,109
1.2.13	February 8, 1995	11MiB	136KiB	4,531
2.0.39	January 9, 2001	35MiB	204KiB	6,792
2.2.22	September 16, 2002	93MiB	292KiB	9,554
2.4.22	August 25, 2003	181MiB	436KiB	15,724
2.6.0-test4	August 22, 2003	261MiB	604KiB	21,714
3.12.2	2013/11/30	518MiB	2.55MiB	
4.4.14	2016/6/25	618MiB	2.88MiB	
4.16	2018/4/1	748MiB	3.64MiB	

4

版本号

- Major.Minor.Revision-Description
- 主版本号.次版本号.修补次数-描述
- 次版本号为偶数时为稳定版本

ge@gewubox:~$ uname -a
Linux gewubox 3.12.2 #3 SMP Sun Mar 5 20:32:24 CST 2017 i686 i686 i386 GNU/Linux

gedu@DESKTOP-4NBEECU:/mnt/c/Windows/System32$ cat /proc/version
Linux version **3.4.0-Microsoft** (Microsoft@Microsoft.com) (gcc version 4.7 (GCC))
#1 SMP PREEMPT Wed Dec 31 14:42:53 PST 2014

Distribution

- Creating a Complete Linux-Based OS
- A Linux kernel
- Core Unix tools
- Supplemental software
- Startup scripts
- An installer

Distribution	Availability	Package format	Release cycle	Administrator skill reqs
Arch	Free	pacman	Rolling	Expert
CentOS	Free	RPM	approx 2-year	Intermediate
Debian	Free	Debian	2-year	Intermediate to expert
Fedora	Free	RPM	approx 6-month	Intermediate
Gentoo	Free	Ebuild	Rolling	Expert
Mint	Free	Debian	6-month	Novice to intermediate
openSUSE	Free	RPM	8-month	Intermediate
Red Hat Enterprise	Commercial	RPM	approx 2-year	Intermediate
Scientific	Free	RPM	approx 6-month	Intermediate to expert
Slackware	Free	tarballs	Irregular	Expert
SUSE Enterprise	Commercial	RPM	2–3 years	Intermediate
Ubuntu	Free	Debian	6-month	Novice to intermediate

Ubuntu

- **Canonical Ltd. 总部在英国**
- **Mark Richard Shuttleworth** (born 18 September 1973) is a South African entrepreneur who is the founder and CEO of Canonical Ltd.
- 右图站立者，第一个进入太空的南非人
- 1995年创建Thawte Consulting，1999被VeriSign收购，赚得5.75亿美金

https://en.wikipedia.org/wiki/Canonical_(company)

THE LINUX FOUNDATION

Founded in 2000, The Linux Foundation today provides tools, training, and events to scale any open source project, which together deliver an economic impact not achievable by any one company.

The World's Most Important Open Source Software Project—Linux

Linux is the world's largest and most pervasive open source software project in history. The Linux Foundation is home to Linux creator Linus Torvalds and lead maintainer Greg Kroah-Hartman and provides a neutral home where Linux kernel development can be protected and accelerated for years to come.

- Of the top one million domains, Linux is the operating system for over 95% of them
- Over 80% of smartphones run Android, which is based on the Linux kernel
- Of the top 500 fastest supercomputers in the world, more than 98% of them run on Linux
- Most of the global markets are running on Linux, including NYSE, NASDAQ, London Exchange, Tokyo Stock Exchange, and more
- The majority of consumer electronic devices use Linux for its small footprint
- More than 75% of cloud-enabled enterprises report using Linux as their primary cloud platform
- Linux is the go-to infrastructure supporting the world's e-commerce leaders, including Amazon, Ebay, Paypal, Walmart, and others

The operating system has seen massive acceptance in almost every sector, including financial, government, education, and even film production. Linux is also the operating system of choice to support cutting-edge technologies such as the Internet of Things, cloud computing, and big data. It is helping to transform industries and disrupt the status quo.

Tainted

- When the kernel is tainted, it means that **it is in a state that is not supported by the community**.

常见污点

- The use of a proprietary (or non-GPL-compatible) kernel module—this is the most common cause of tainted kernels and usually results from loading proprietary NVIDIA or AMD video drivers
- The use of staging drivers, which are part of the kernel source code but are not fully tested
- The use of out-of-tree modules that are not included with the Linux kernel source code
- Forcible loading or unloading of a kernel module (such as forcibly inserting a module not built for the current version of the kernel)
- The use of an SMP (multiprocessor) kernel on certain unsupported uniprocessor CPUs, primarily older AMD Athlon processors
- Overriding of the ACPI DSDT, sometimes needed to correct for power-management bugs (seehere for details)
- Certain critical error conditions, such as machine check exceptions and kernel oopses
- Certain serious bugs in the system firmware (BIOS, UEFI) which the kernel must work around

Case Study

[0.923328] nvidia: loading out-of-tree module taints kernel.
[0.923330] nvidia: module license 'NVIDIA' taints kernel.
[0.923331] Disabling lock debugging due to kernel taint

LINUX CREATOR LINUS Torvalds may call Nvidia "the single worst company" the Linux community has ever dealt with. But the chipmaker makes no apologies for its approach to the open source operating system.

13

TALK IS CHEAP. SHOW ME THE CODE.

Directory	Description
arch	Architecture-specific source
block	Block I/O layer
crypto	Crypto API
Documentation	Kernel source documentation
drivers	Device drivers
firmware	Device firmware needed to use certain drivers
fs	The VFS and the individual filesystems
include	Kernel headers
init	Kernel boot and initialization
ipc	Interprocess communication code
kernel	Core subsystems, such as the scheduler
lib	Helper routines
mm	Memory management subsystem and the VM
net	Networking subsystem
samples	Sample, demonstrative code
scripts	Scripts used to build the kernel
security	Linux Security Module
sound	Sound subsystem
usr	Early user-space code (called initramfs)
tools	Tools helpful for developing Linux
virt	Virtualization infrastructure

14

files in the root

.gitignore	1,097	08/13/2016 15:44	-a--c
.mailmap	4,465	08/13/2016 15:44	-a--c
Copying	18,693	08/13/2016 15:44	-a--c
Credits	95,317	08/13/2016 15:44	-a--c
Kbuild	2,536	08/13/2016 15:44	-a--c
Kconfig	252	08/13/2016 15:44	-a--c
MAINTAINERS	260,046	08/13/2016 15:44	-a--c
Makefile	48,517	08/13/2016 15:44	-a--c
Readme	18,736	08/13/2016 15:44	-a--c
REPORTING-BUGS	7,485	08/13/2016 15:44	-a--c

COPYING

- the kernel license (the GNU GPL v2).

CREDITS

- listing of developers with more than a trivial amount of code in the kernel.

MAINTAINERS

- the names of the individuals who maintain subsystems and drivers in the kernel.

Makefile

- the base kernel Makefile.

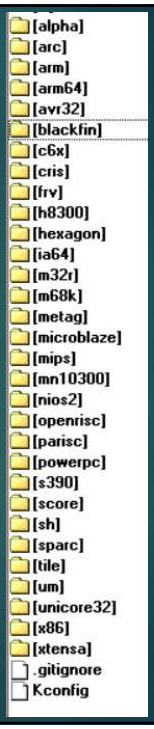
CREDITS

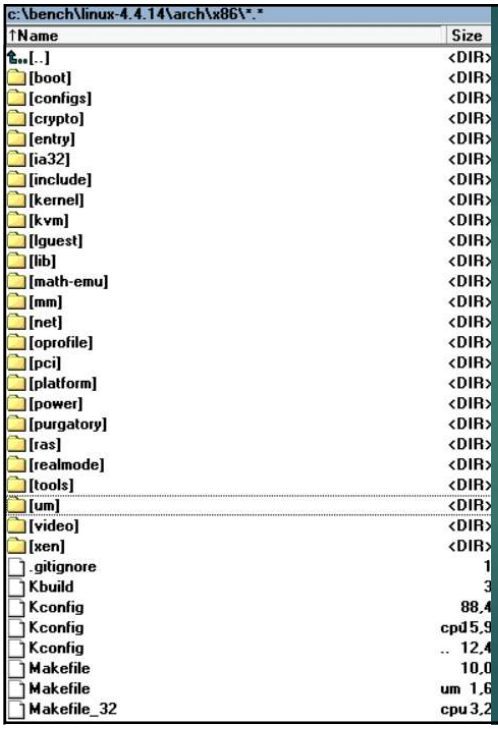
```
        This is at least a partial credits-file of people that have
        contributed to the Linux project.  It is sorted by name and
        formatted to allow easy grepping and beautification by
        scripts.  The fields are: name (N), email (E), web-address
        (W), PGP key ID and fingerprint (P), description (D), and
        snail-mail address (S).
        Thanks,

                        Linus
----------
```

```
N: Linus Torvalds
E: torvalds@linux-foundation.org
D: Original kernel hacker
S: Portland, Oregon 97005
S: USA
```

arch/

- 针对特定CPU架构的代码
- 每种架构1/2（32位/64位）个子目录
- 体量远远超过kernel目录
  - 91MB : 6.1 MB (4.4)





x86

- 与一级子目录对应的架构相关部分
  - /include
  - /mm
  - /kernel
  - /lib
  - /net
- 虚拟化支持
  - /kvm
  - /xen
  - /math-emu
  - /lguest

kernel

- 内核的‘核心’
- 中断管理
- 进程管理器
  - 任务调度(sched子目录)
- 调试支持
  - Ptrace，kdb，kgdb
  - ftrace
- 启动

19

根目录的文件

.gitignore	acct.c	async.c	audit.c	audit.h
auditfilter.c	auditsc.c	audit_fsnotify.c	audit_tree.c	audit_watch.c
backtracetest.c	bounds.c	capability.c	cgroup.c	cgroup_freezer.c
cgroup_pids.c	compat.c	configs.c	context_tracking.c	cpu.c
cpuset.c	cpu_pm.c	crash_dump.c	cred.c	delayacct.c
dma.c	elfcore.c	exec_domain.c	exit.c	extable.c
fork.c	freezer.c	futex.c	futex_compat.c	groups.c
hung_task.c	irq_work.c	jump_label.c	kallsyms.c	kcmp.c
Kconfig.freezer	Kconfig.hz	Kconfig.locks	Kconfig.preempt	kexec.c
kexec_core.c	kexec_file.c	kexec_internal.h	kmod.c	kprobes.c
ksysfs.c	kthread.c	latencytop.c	Makefile	membarrier.c
memremap.c	module-internal.h	module.c	module_signing.c	notifier.c
nsproxy.c	padata.c	panic.c	params.c	pid.c
pid_namespace.c	profile.c	ptrace.c	range.c	reboot.c
relay.c	resource.c	seccomp.c	signal.c	smp.c
smpboot.c	smpboot.h	softirq.c	stacktrace.c	stop_machine.c
sys.c	sysctl.c	sysctl_binary.c	sys_ni.c	taskstats.c
task_work.c	test_kprobes.c	torture.c	tracepoint.c	tsacct.c
uid16.c	up.c	user-return-notifier.c	user.c	user_namespace.c
utsname.c	utsname_sysctl.c	watchdog.c	workqueue.c	workqueue_internal.h

20

子目录

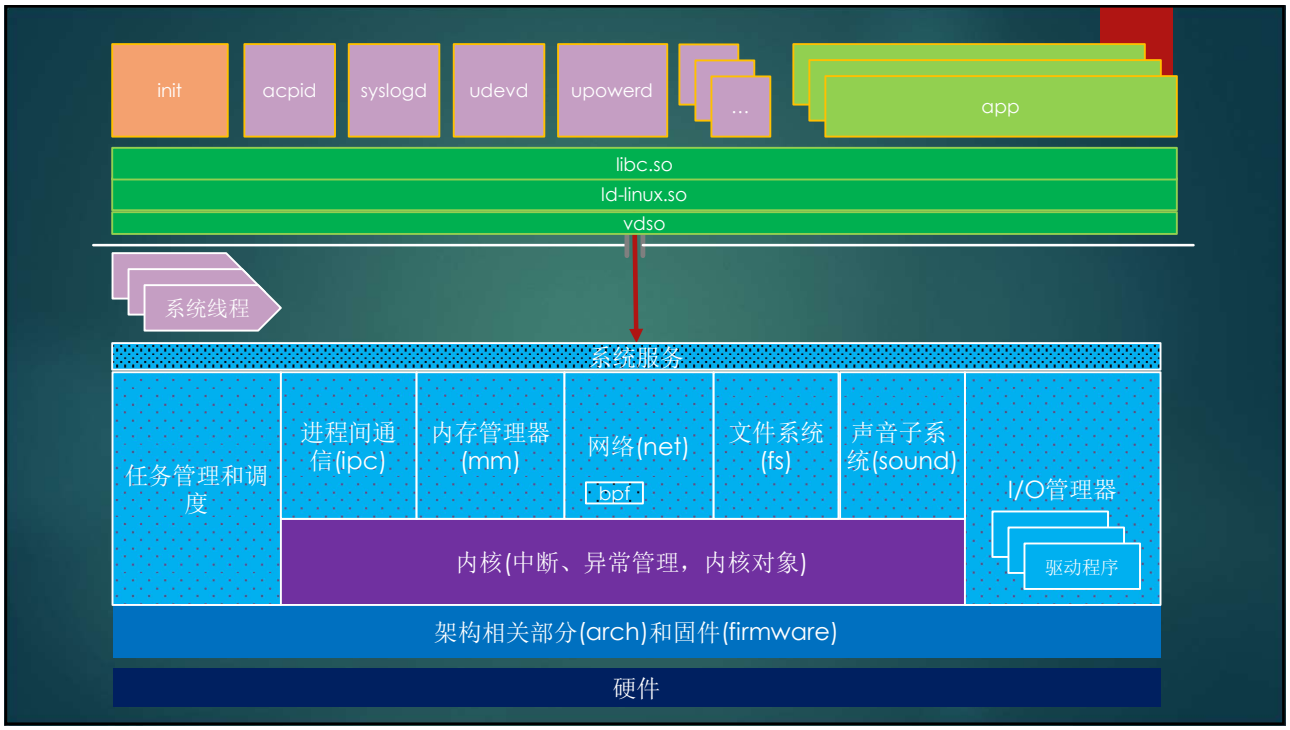
- [bpf] 基于Berkeley Packet Filter的内核态网络包过滤
- [configs] 编译选项
- [debug] 内核调试支持
- [events] 性能监视和事件追踪
- [gcov] 与GCC配合的优化设施(profiling)
- [irq] 中断管理，CPU间通信
- [livepatch] 热修补

续

- [locking] 同步对象
- [power] 电能管理，睡眠
- [printk] 打印输出
- [rcu] RCU(Read-Copy Update)方式的同步机制
- [sched] 任务调度
- [time] 时钟和计时器(timer)
- [trace] 函数追踪


版本简史
源码目录
架构
内核参数
文档和工具链

23


init
acpid
syslogd
udevd
upowerd
...
app
libc.so
ld-linux.so
vdso
系统线程
系统服务
任务管理和调度
进程间通信(ipc)
内存管理器(mm)
网络(net)
bpf
文件系统(fs)
声音子系统(sound)
I/O管理器
驱动程序
内核(中断、异常管理，内核对象)
架构相关部分(arch)和固件(firmware)
硬件

24

Linux Architecture

- Monolithic kernel ('单板'内核)
  - Contains modular components, however
- UNIX-like or UNIX-based operating system
- Six primary subsystems:
  - Process management
  - Interprocess communication
  - Memory management
  - File system management
    - VFS: provides a single interface to multiple file systems
  - I/O management
  - Networking

25

ld

- Loader
- Ld-linux.so.2
- ld-2.15.so

```
ge@gewubox:~/eglibc-2.15/elf$ ll /lib/ld-linux.so.2
lrwxrwxrwx 1 root root 25 Mar 24 04:33 /lib/ld-linux.so.2 -> i386-linux-gnu/ld-2.15.so*
```

26

```
(gdb) info shared
From        To          Syms Read   Shared Object Library
0xb7fde820  0xb7ff6b5f  Yes         /lib/ld-linux.so.2
(gdb) bt
#0  __brk (addr=0x0) at ../sysdeps/unix/sysv/linux/i386/brk.c:36
#1  0xb7ff28ce in frob_brk () at ../sysdeps/unix/sysv/linux/dl-sysdep.c:36
#2  _dl_sysdep_start (start_argptr=0xbffff400, dl_main=0xb7fdff90 <dl_main>)
    at ../elf/dl-sysdep.c:218
#3  0xb7fe2eab in _dl_start_final (arg=0xbffff400) at rtld.c:337
#4  _dl_start (arg=0xbffff400) at rtld.c:563
#5  0xb7fdf1d7 in _start () from /lib/ld-linux.so.2
```

用户态执行起点

源代码

- eglibc/elf
- Dl-xxx
- Dynamic loader

```
ge@gewubox:~/eglibc-2.15/elf$ ls dl*
dl-addr.c      dl-debug.c      dl-fini.c         dl-lookup.c   dl-profile.c   dl-support.c     dl-tsd.c
dl-brk.c       dl-deps.c       dl-fptr.c         dl-minimal.c  dl-profstub.c  dl-symaddr.c     dl-version.c
dl-cache.c     dl-dst.h        dl-init.c         dl-misc.c     dl-reloc.c     dl-sym.c
dl-caller.c    dl-environ.c    dl-iteratephdr.c  dl-object.c   dl-runtime.c   dl-sysdep.c
dl-close.c     dl-error.c      dl-libc.c         dl-open.c     dl-sbrk.c      dl-tls.c
dl-conflict.c  dl-execstack.c  dl-load.c         dl-origin.c   dl-scope.c     dl-trampoline.c
```

elf/rtld.c

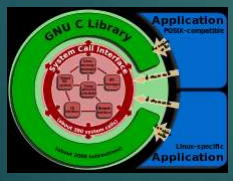
- Run time dynamic linker.

NOTES ✖ FAQ ✖ rtld.c ✖

```
static ElfW(Addr) __attribute_used__ internal_function
_dl_start (void *arg)
{
#ifdef DONT_USE_BOOTSTRAP_MAP
# define bootstrap_map GL(dl_rtld_map)
#else
  struct dl_start_final_info info;
# define bootstrap_map info.l
#endif

  /* This #define produces dynamic linking inline functions for
     bootstrap relocation instead of general-purpose relocation.
     Since ld.so must not have any undefined symbols the result
     is trivial: always the map of ld.so itself.  */
#define RTLD_BOOTSTRAP
#define RESOLVE_MAP(sym, version, flags) (&bootstrap_map)
#include "dynamic-link.h"

  if (HP_TIMING_INLINE && HP_TIMING_AVAIL)
#ifdef DONT_USE_BOOTSTRAP_MAP
    HP_TIMING_NOW (start_time);
#else
    HP_TIMING_NOW (info.start_time);
#endif

  /* Partly clean the `bootstrap_map' structure up.  Don't use
     `memset' since it might not be built in or inlined and we cannot
     make function calls at this point.  Use '__builtin_memset' if we
     know it is available.  We do not have to clear the memory if we
     do not have to use the temporary bootstrap_map.  Global variables
     are initialized to zero by default.  */
#ifndef DONT_USE_BOOTSTRAP_MAP
# ifdef HAVE_BUILTIN_MEMSET
  __builtin_memset (bootstrap_map.l_info, '\0', sizeof (bootstrap_map.l_info));
# else
  for (size_t cnt = 0;
       cnt < sizeof (bootstrap_map.l_info) / sizeof (bootstrap_map.l_info[0]);
       ++cnt)
    bootstrap_map.l_info[cnt] = 0;
# endif
# if USE___THREAD
  bootstrap_map.l_tls_modid = 0;
# endif
#endif
```


glibc

- GNU C Library
- LINUX程序的‘中坚’力量
- 始于1987年



The GNU C Library (glibc)

What is glibc? | Get Started | Get Involved | Sources | Documentation | Report a bug

What is glibc?

The GNU C Library project provides *the* core libraries for the GNU system and GNU/Linux systems, as well as many other systems that use Linux as the kernel. These libraries provide critical APIs including ISO C11, POSIX.1-2008, BSD, OS-specific APIs and more. These APIs include such foundational facilities as `open`, `read`, `write`, `malloc`, `printf`, `getaddrinfo`, `dlopen`, `pthread_create`, `crypt`, `login`, `exit` and more.

The GNU C Library is designed to be a backwards compatible, portable, and high performance ISO C library. It aims to follow all relevant standards including ISO C11, POSIX.1-2008, and IEEE 754-2008.

The project was started circa 1988 and is almost 30 years old. You can see the complete project release history on the wiki.

Despite the project's age there is still a lot to do so please Get Started and Get Involved!

Current Status

The GNU C Library releases every 6 months. See the NEWS file in the glibc sources for more information.

- The current stable version of glibc is 2.25, released on February 5, 2017.
- The current development version of glibc 2.26, releasing on or around August 1, 2017.

Latest News

2017-02-05: glibc 2.25 released.

2016-11-24: Test results mailing list is active.

2016-08-05: glibc 2.24 released.

2016-02-19: glibc 2.23 released.

2016-02-16: CVE-2015-7547: glibc getaddrinfo() stack-based buffer overflow -- Fixed on development branch for glibc 2.23 release.

GNU项目

- Richard Stallman创立于1983年9月
- 旨在建立完全由自由软件组成的类Unix操作系统
- GNU General Public License (GPL)

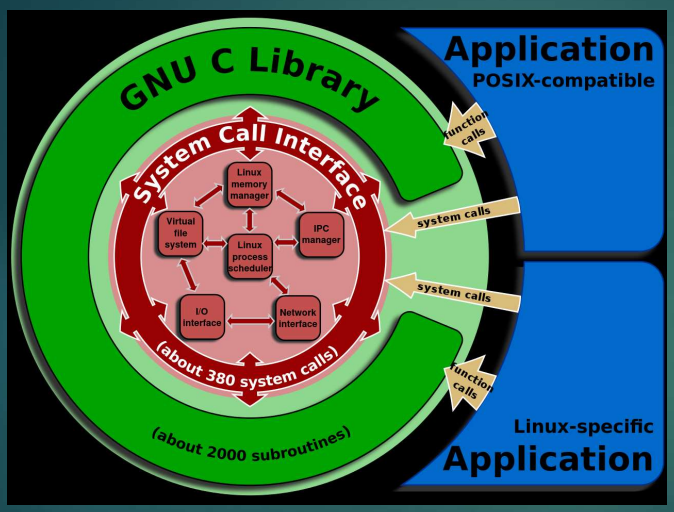
31

GNU Compiler Collection

- GNU工具链的核心部件
- 最初版本发布于1987年
- 8.0 / April 20, 2017

- “GNU工程开始不久，我听说了自由大学编译器工具包，又叫VUCK。（荷兰语的“自由”一词以v开头。）这是一个为处理多种语言而设计的编译器，它可以处理C语言和Pascal语言，还可以支持多个目标计算机。我写信问其作者GNU是否可以使用该工具包。他带着嘲弄的口吻回答，大学是自由的，但编译器不是。我因此决定我的第一个GNU程序就是做一个支持多语言、多平台的编译器。”
  - GNU工程，Richard Stallman
  - https://www.gnu.org/gnu/thegnuproject.html

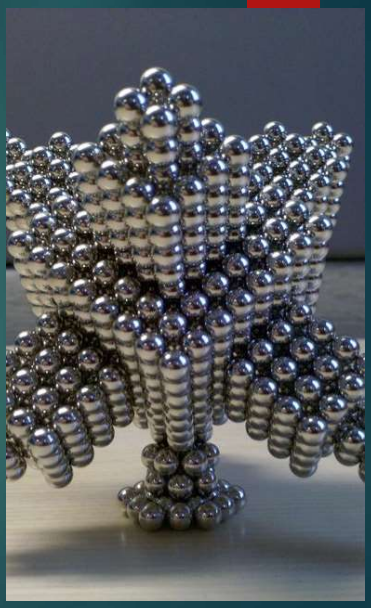
32



沟通用户空间和内核空间的桥梁

Roland McGrath

- Glibc 奠基者
  - 始于1987
- http://www.frob.com/~roland/



Roland McGrath bows out as glibc maintainer

From: Roland McGrath <roland-AT-hack.frob.com>
To: "GNU C. Library" <libc-alpha-AT-sourceware.org>
Subject: I'm already gone
Date: Fri, 7 Jul 2017 00:21:12 -0700 (PDT)
Message-ID: <20170707072112.1272F2C3BAA@topped-with-meat.com>
Archive-link: Article

Hello, friends!

You might have noticed that I haven't been present on the list or perhaps answered your direct email in several months. I'm sorry I've been away so long without a word, but I'm not coming back any time soon. There's no big news with me. I've just found that I've drifted away and today I'm acknowledging what's already happened.

This summer marks 30 years since I began writing the GNU C Library. (That's two thirds of my lifespan so far.) It's long enough.

So, I'm hereby declaring myself maintainer emeritus and withdrawing from direct involvement in the project. These past several months, if not the last few years, have proven that you don't need me any more.

You'll make good decisions, as you've already made good decisions. You'll actually get around to implementing some of the things I've been suggesting or meaning to do (or saying I would do) for years, as you've already made progress on some of those ideas in recent months. If I stayed around to give advice, you'd ignore my advice to be more paranoid and more cautious, plow ahead anyway, ship it, and then have to redress the problem when the practical issues manifested, as you've already done and had to do. :-) All in all, I have no doubt at all that the job you're doing now and will do in the future maintaining glibc is better than I ever did that job myself and at least as good as my presence in the project might ever make it.

https://lwn.net/Articles/727383/

Software 30

Roland McGrath steps down as glibc maintainer after 30 years

Open source is only for two-thirds of a life, says veteran

By Richard Chirgwin 10 Jul 2017 at 02:58 SHARE ▼

Open source luminary Roland McGrath has decided "enough is enough" – after 30 years on the GNU compiler library project.

As a teenager in 1987 – working back from the age he gives in his mailing list post, as a 15-year-old, in fact – McGrath began writing glibc, and he reckons that devoting "two thirds of my lifespan so far" is "long enough".

The library's purpose was to deliver the functionality required in an ANSI-standard C library, and the Free Software Foundation said that mission was "nearly complete" in February 1988.

"So, I'm hereby declaring myself maintainer emeritus and withdrawing from direct involvement in the project. These past several months, if not the last few years, have proven that you don't need me any more", he continues.

Some individuals, he says, have contributed more to the project than McGrath, and the collective effort has done "far more than I ever could have".

"I'm especially grateful to the small handful of folks who contributed in the early days when so much was so different than it is today; to the diehard few who've hung on through all the changes and tribulations over the many years; and to those, old and new, who have come together in recent years to breathe new life into the project and steer us towards

https://www.theregister.co.uk/2017/07/10/glibc_maintainer_roland_mcgrath_steps_down/

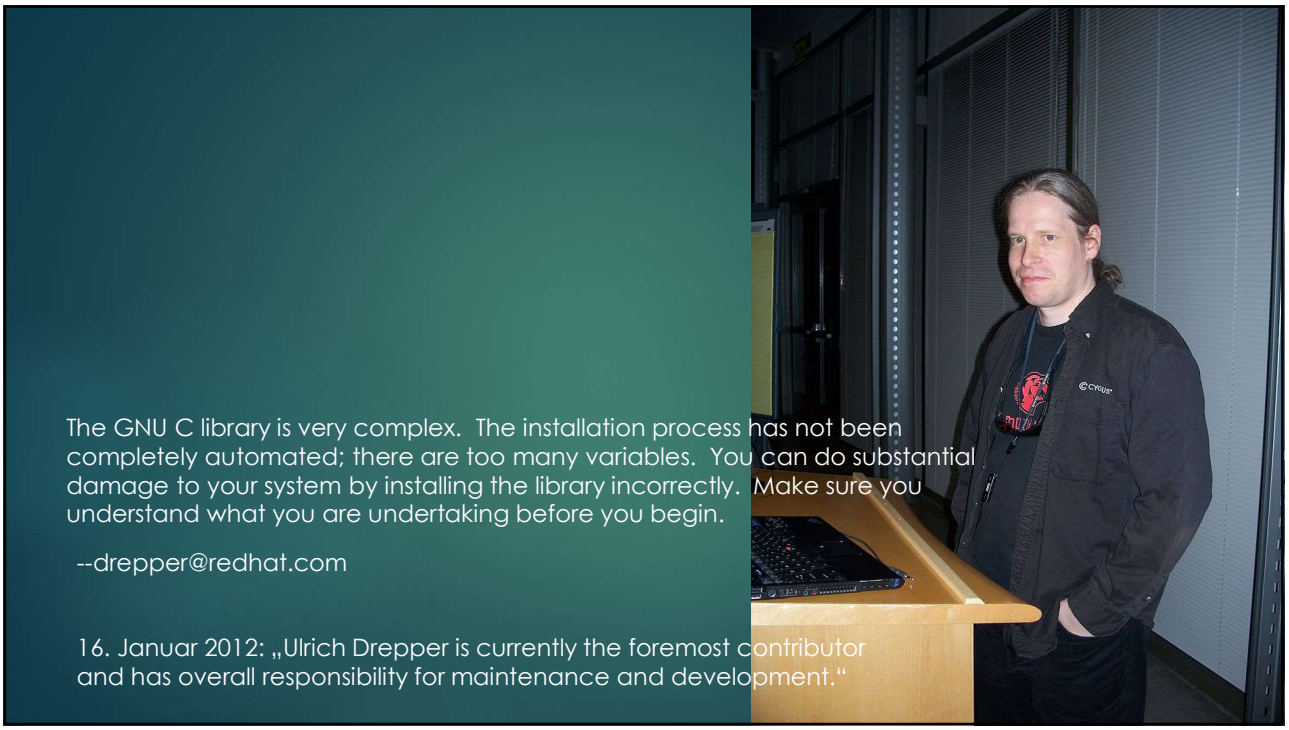
ChangeLog.15 | ChangeLog.5 | ChangeLog.9 | ChangeLog.1

```
Thu Sep  3 17:31:13 1992  Roland McGrath  (roland@geech.gnu.ai.mit.edu)

    * sysdeps/unix/bsd/setgroups.S: New file.  How did this manage not
    to exist already??

    * Version 1.04.

    * sysdeps/generic/memmem.c (memmem): Fixed loop condition not to
    use nonexistent variable.

    * string/string.h (memmem): Put const qualifier on args.

    * sysdeps/stub/sigaltstack.c (sigaltstack): Fix arg type.

    * setjmp/sigsetjmp.c: #undef sigsetjmp before defining the function.

Wed Sep  2 16:43:58 1992  Roland McGrath  (roland@geech.gnu.ai.mit.edu)

    * misc/Makefile (headers): Add syscall.h.
    * sysdeps/unix/Makefile: Only generate syscall.h if it would
    otherwise come from stub.
    * sysdeps/stub/syscall.h: New file.

    * time/Makefile (routines): Add stime.

    * io/Makefile (routines): Add fchdir.

    * signal/Makefile (routines): Add sigaltstack.

    * string/Makefile (routines): Add memmem.

    * setjmp/Makefile (routines): Add sigsetjmp, _setjmp.

    * misc/Makefile (routines): Add getpass.

    * Makefile (distribute): Add NEWS.

    * Makerules (ar-it) [! objdir]: Pass `ru' instead of `u' to ar.
    Use $(..)libc.a instead of $(libc.a).

    * sysdeps/unix/sysv/i386/linux/sysdep.h (PSEUDO): Call numbers are
    SYS_*, not __NR_*.

    * sysdeps/unix/sysv/i386/linux/__wait.S: Prepend extra _ to
    `__waitpid' for jmp.
```

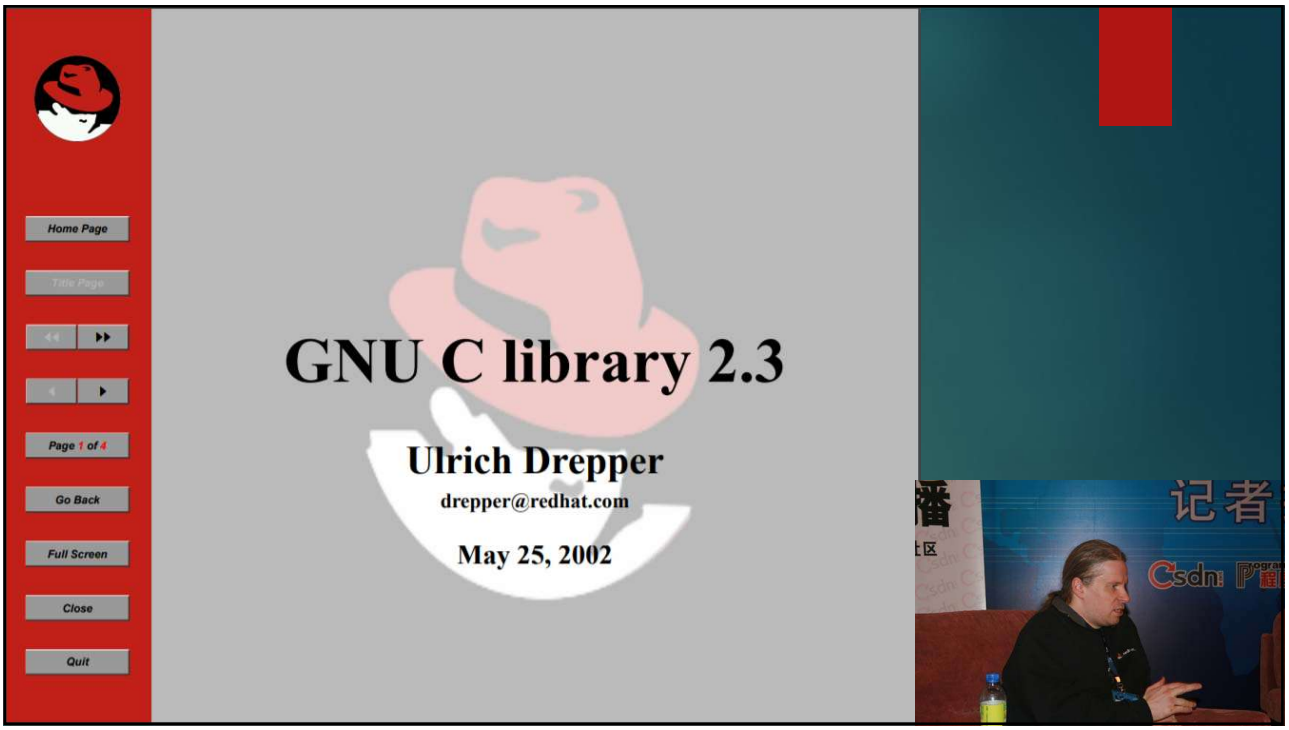
20年

FAQ | rtld.c | README | read.c | utime.c | ChangeLog

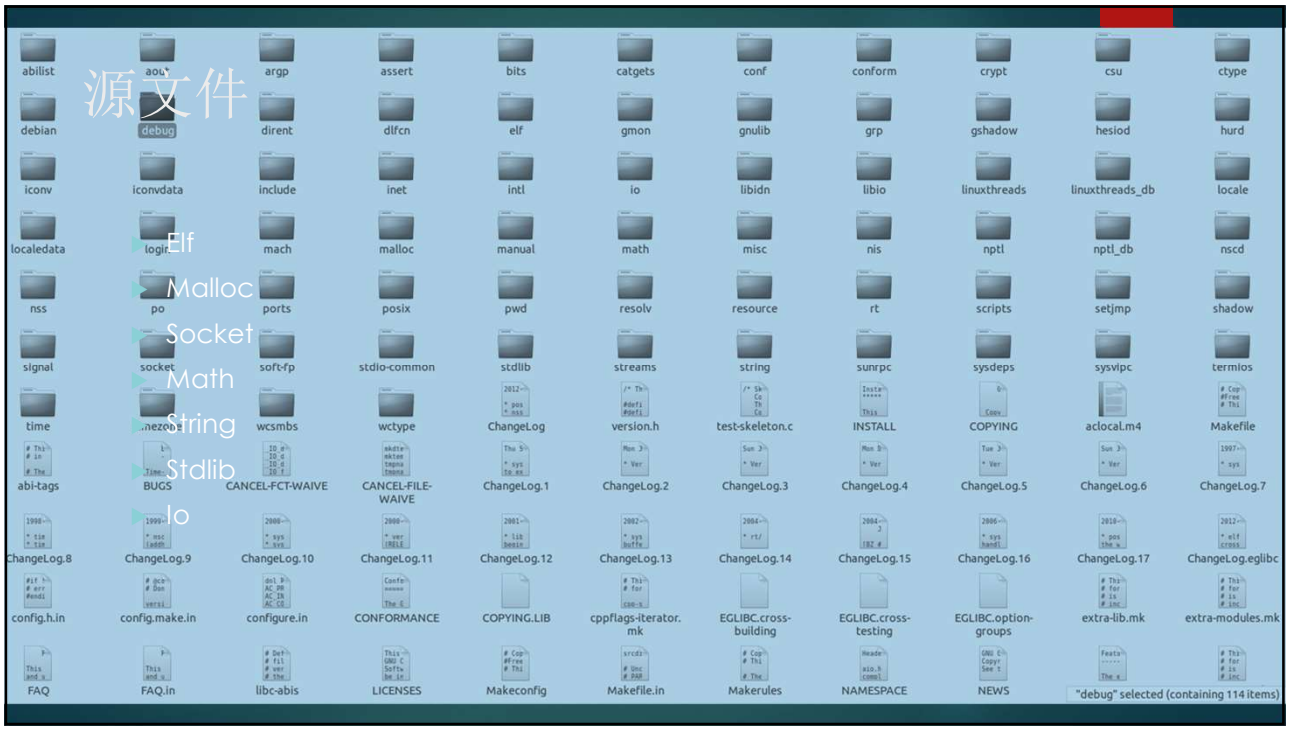
```
2012-01-01  Ulrich Drepper  <drepper@gmail.com>

    * posix/getconf.c: Update copyright year.
    * nss/getent.c: Likewise.
    * nss/makedb.c: Likewise.
    * iconv/iconvconfig.c: Likewise.
    * iconv/iconv_prog.c: Likewise.
    * elf/ldconfig.c: Likewise.
    * elf/pldd.c: Likewise.
    * elf/sotruss.ksh: Likewise.
    * catgets/gencat.c: Likewise.
    * csu/version.c: Likewise.
    * elf/ldd.bash.in: Likewise.
    * elf/sprof.c (print_version): Likewise.
    * locale/programs/locale.c: Likewise.
    * locale/programs/localedef.c: Likewise.
    * login/programs/pt_chown.c: Likewise.
    * nscd/nscd.c (print_version): Likewise.
    * debug/xtrace.sh: Likewise.
    * malloc/memusage.sh: Likewise.
    * malloc/mtrace.pl: Likewise.
    * debug/catchsegv.sh: Likewise.

    2-30  Jakub Jelinek  <jakub@redhat.com>

    posix/regex_internal.c (re_string_fetch_byte_case): Remove
    re attribute.

2011-12-23  Ulrich Drepper  <drepper@gmail.com>

    * version.h (RELEASE): Bump for 2.15 release.
    * include/features.h (__GLIBC_MINOR__): Bump to 15.

    * sysdeps/x86_64/dl-machine.h: Fix typos in comments.
    Patch by Marek Polacek <mpolacek@redhat.com>.

    * bits/byteswap.h: Protect long long constants with __extension__.
    * sysdeps/i386/bits/byteswap.h: Likewise.
    * sysdeps/ia64/bits/byteswap.h: Likewise.
    * sysdeps/s390/bits/byteswap.h: Likewise.
    * sysdeps/x86_64/bits/byteswap.h: Likewise.

2011-12-23  Liubov Dmitrieva  <liubov.dmitrieva@gmail.com>

    [BZ #13540]
```


The GNU C library is very complex. The installation process has not been completely automated; there are too many variables. You can do substantial damage to your system by installing the library incorrectly. Make sure you understand what you are undertaking before you begin.
--drepper@redhat.com
16. Januar 2012: „Ulrich Drepper is currently the foremost contributor and has overall responsibility for maintenance and development."

37


Home Page
Title Page
Page 1 of 4
Go Back
Full Screen
Close
Quit
GNU C library 2.3
Ulrich Drepper
drepper@redhat.com
May 25, 2002

38



调用内核服务

40

- 从用户态切换到内核态
- IA32 CPU设计了专用的指令(从奔腾II开始)
  - SYSENTER
  - SYSEXIT
- 或者使用INT 2E(NT) INT 80 (LINUX)

MSR寄存器名称	用途
SYSENTER_CS_MSR	目标代码段的CS选择子
SYSENTER_ESP_MSR	目标ESP
SYSENTER_EIP_MSR	目标EIP

只能从固定的入口（gate）进入内核

VDSO

- virtual dynamic shared object
- mapped into every process address space in Linux since kernel version 2.6.x
- contains code to speed up system calls, and they can be invoked directly from the VDSO

41

```
ge@gewubox:~/work/llaolao3$ cat /proc/self/maps
08048000-08053000 r-xp 00000000 08:01 393235     /bin/cat
08053000-08054000 r--p 0000a000 08:01 393235     /bin/cat
08054000-08055000 rw-p 0000b000 08:01 393235     /bin/cat
081a7000-081c8000 rw-p 00000000 00:00 0          [heap]
b737d000-b757d000 r--p 00000000 08:01 268965     /usr/lib/locale/locale-archive
b757d000-b757e000 rw-p 00000000 00:00 0
b757e000-b7722000 r-xp 00000000 08:01 132073     /lib/i386-linux-gnu/libc-2.15.so
b7722000-b7724000 r--p 001a4000 08:01 132073     /lib/i386-linux-gnu/libc-2.15.so
b7724000-b7725000 rw-p 001a6000 08:01 132073     /lib/i386-linux-gnu/libc-2.15.so
b7725000-b7728000 rw-p 00000000 00:00 0
b7737000-b7738000 r--p 005e0000 08:01 268965     /usr/lib/locale/locale-archive
b7738000-b773a000 rw-p 00000000 00:00 0
b773a000-b773b000 r-xp 00000000 00:00 0          [vdso]
b773b000-b775b000 r-xp 00000000 08:01 132053     /lib/i386-linux-gnu/ld-2.15.so
b775b000-b775c000 r--p 0001f000 08:01 132053     /lib/i386-linux-gnu/ld-2.15.so
b775c000-b775d000 rw-p 00020000 08:01 132053     /lib/i386-linux-gnu/ld-2.15.so
bfb15000-bfb36000 rw-p 00000000 00:00 0          [stack]
```

42

```
ge@gewubox:~$ sudo grep "vdso" /proc/kallsyms
[sudo] password for ge:
c191b63c r __ksymtab_vdso_enabled
c1922e8c r __kcrctab_vdso_enabled
c1924216 r __kstrtab_vdso_enabled
c19dffc0 D vdso_enabled
c19feac1 t vdso_setup
c1a545a8 T vdso32_int80_start
c1a54c38 T vdso32_int80_end
c1a54c38 T vdso32_syscall_end
c1a54c38 T vdso32_syscall_start
c1a54c38 T vdso32_sysenter_start
c1a552ac T vdso32_sysenter_end
c1a9825d t __setup_str_vdso32_setup
c1a98263 t __setup_str_vdso_setup
c1aa37cc t __setup_vdso32_setup
c1aa37d8 t __setup_vdso_setup
c1ad8e60 b vdso32_pages
c1ad8edc b vdso_mapped.19946
```

43

```
(gdb) bt
#0  vfs_read (file=0xd1892a80,
    buf=0xb76f0000 <Address 0xb76f0000 out of bounds>, count=1024,
    pos=0xd189bf98) at fs/read_write.c:382
#1  0xc1178a57 in SYSC_read (count=1024,
    buf=0xb76f0000 <Address 0xb76f0000 out of bounds>, fd=<optimized out>)
    at fs/read_write.c:506
#2  SyS_read (fd=8, buf=-1217462272, count=1024) at fs/read_write.c:499
#3  <signal handler called>
#4  0xb76f3424 in ?? ()
#5  0xb76f0000 in ?? ()
#6  0xb76f0000 in ?? ()
#7  0xb76f0000 in ?? ()
#8  0xb76f0000 in ?? ()
#9  0xb76f0000 in ?? ()
#10 0xb76f0000 in ?? ()
#11 0xb76f0000 in ?? ()
#12 0xb76f0400 in ?? ()
#13 0x00000000 in ?? ()
```

44

```
/*
 *
 *  Copyright (C) 1991, 1992  Linus Torvalds
 */

/*
 * entry.S contains the system-call and fault low-level handling routines.
 * This also contains the timer-interrupt handler, as well as all interrupts
 * and faults that can result in a task-switch.
 *
 * NOTE: This code handles signal-recognition, which happens every time
 * after a timer-interrupt and after each system call.
 *
 * I changed all the .align's to 4 (16 byte alignment), as that's faster
 * on a 486.
 *
 * Stack layout in 'syscall_exit':
 *       ptrace needs to have all regs on the stack.
 *       if the order here is changed, it needs to be
 *       updated in fork.c:copy_process, signal.c:do_signal,
 *       ptrace.c and ptrace.h
 *
```

```
kd>kn
 # ChildEBP RetAddr
00 d7eaff88 c1178a57 uk!vfs_read [fs/read_write.c @ 382]
01 d7eaffac c16830cd uk!SyS_read+0x57 [fs/read_write.c @ 507]
02 d7eae000 c1962360 uk!ia32_sysenter_target+0x89 [arch/x86/kernel/entry_32.S @ 439]
```

45

```
    # sysenter call handler stub
ENTRY(ia32_sysenter_target)
    CFI_STARTPROC simple
    CFI_SIGNAL_FRAME
    CFI_DEF_CFA esp, 0
    CFI_REGISTER esp, ebp
    movl TSS_sysenter_sp0(%esp),%esp
```

7.10.21 .cfi_signal_frame

Mark current function as signal trampoline.

```
00005510 00000010 ffffffff CIE
  Version:               1
  Augmentation:          "S"
  Code alignment factor: 1
  Data alignment factor: -4
  Return address column: 8

  DW_CFA_def_cfa: r4 (esp) ofs 0
  DW_CFA_register: r4 (esp) in r5 (ebp)

00005524 00000058 00005510 FDE cie=00005510 pc=c1683044..c1683162
  DW_CFA_advance_loc: 9 to c168304d
  DW_CFA_def_cfa_offset: 4
  DW_CFA_advance_loc: 1 to c168304e
  DW_CFA_def_cfa_offset: 8
  DW_CFA_offset: r4 (esp) at cfa-8
  DW_CFA_advance_loc: 1 to c168304f
  DW_CFA_def_cfa_offset: 12
  DW_CFA_advance_loc: 9 to c1683058
  DW_CFA_def_cfa_offset: 16
  DW_CFA_advance_loc: 7 to c168305f
  DW_CFA_def_cfa_offset: 20
  DW_CFA_offset: r8 (eip) at cfa-20
  DW_CFA_advance_loc: 1 to c1683060
  DW_CFA_def_cfa_offset: 24
  DW_CFA_advance_loc: 3 to c1683063
  DW_CFA_def_cfa_offset: 28
  DW_CFA_advance_loc: 2 to c1683065
  DW_CFA_def_cfa_offset: 32
  DW_CFA_advance_loc: 1 to c1683066
  DW_CFA_def_cfa_offset: 36
  DW_CFA_advance_loc: 1 to c1683067
  DW_CFA_def_cfa_offset: 40
  DW_CFA_advance_loc: 1 to c1683068
  DW_CFA_def_cfa_offset: 44
```

46

```
kd>k
ChildEBP RetAddr
d7eaff88 c1178a57 uk!vfs_read [fs/read_write.c @ 382]
d7eaffac c16830cd uk!SyS_read+0x57 [fs/read_write.c @ 507]
d7eafff4 b7794424 uk!ia32_sysenter_target+0x89 [arch/x86/kernel/entry_32.S @ 439]

kd>u b779441f
b779441f 90          nop
b7794420 90          nop
b7794421 90          nop
b7794422 cd80        int    80h
b7794424 5d          pop    ebp
b7794425 5a          pop    edx
b7794426 59          pop    ecx
b7794427 c3          ret
```

b7794000-b7795000 r-xp 00000000 00:00 0 [vdso]

47

```
        CFI_REL_OFFSET          ecx, 0
        pushl   %edx
        CFI_ADJUST_CFA_OFFSET   4
        CFI_REL_OFFSET          edx, 0
        pushl   %ebp
        CFI_ADJUST_CFA_OFFSET   4
        CFI_REL_OFFSET          ebp, 0

        #define SYSENTER_SEQUENCE       "movl %esp, %ebp; sysenter"
        #define SYSCALL_SEQUENCE        "movl %ecx, %ebp; syscall"

#ifdef CONFIG_X86_64
        /* If SYSENTER (Intel) or SYSCALL32 (AMD) is available, use it. */
        ALTERNATIVE_2 "", SYSENTER_SEQUENCE, X86_FEATURE_SYSENTER32, \
                          SYSCALL_SEQUENCE,  X86_FEATURE_SYSCALL32
#else
        ALTERNATIVE "", SYSENTER_SEQUENCE, X86_FEATURE_SEP
#endif

        /* Enter using int $0x80 */
        int     $0x80
GLOBAL(int80_landing_pad)
```

d:\bench\linux-4.4.14\arch\x86\entry\vdso\vdso32*.*

Name	Ext	Size	Date	Attr
[..]		<DIR>	06/25/2016 01:18	——
.gitignore		11	06/25/2016 01:18	-a—
note	S	1,676	06/25/2016 01:18	-a—
sigreturn	S	5,139	06/25/2016 01:18	-a—
system_call	S	2,460	06/25/2016 01:18	-a—
vclock_gettime	c	618	06/25/2016 01:18	-a—
vdso32.lds	S	637	06/25/2016 01:18	-a—
vdso-fakesections	c	34	06/25/2016 01:18	-a—

48

49

Kernel Parameters

The following is a consolidated list of the kernel parameters as implemented by the __setup(), core_param() and module_param() macros and sorted into English Dictionary order (defined as ignoring all punctuation and sorting digits before letters in a case insensitive manner), and with descriptions where known.

The kernel parses parameters from the kernel command line up to "--"; if it doesn't recognize a parameter and it doesn't contain a '.', the parameter gets passed to init: parameters with '=' go into init's environment, others are passed as command line arguments to init. Everything after "--" is passed as an argument to init.

Module parameters can be specified in two ways: via the kernel command line with a module name prefix, or via modprobe, e.g.:

```
        (kernel command line) usbcore.blinkenlights=1
        (modprobe command line) modprobe usbcore blinkenlights=1
```

Parameters for modules which are built into the kernel need to be specified on the kernel command line. modprobe looks through the kernel command line (/proc/cmdline) and collects module parameters when it loads a module, so the kernel command line can be used for loadable modules too.

Hyphens (dashes) and underscores are equivalent in parameter names, so

```
        log_buf_len=1M print-fatal-signals=1
```

can also be entered as

```
        log-buf-len=1M print_fatal_signals=1
```

Double-quotes can be used to protect spaces in values, e.g.:

```
        param="spaces in here"
```

This document may not be entirely up to date and comprehensive. The command "modinfo -p ${modulename}" shows a current list of all parameters of a loadable module. Loadable modules, after being loaded into the running kernel, also reveal their parameters in /sys/module/${modulename}/parameters/. Some of these parameters may be changed at runtime by the command "echo -n ${value} > /sys/module/${modulename}/parameters/${parm}".

The parameters listed below are only valid if certain kernel build options were enabled and if respective hardware is present. The text in square brackets at the beginning of each description states the restrictions within which a parameter is applicable:

内核参数

- 运行期定制
  - 相对于编译期

50

console=

- ttyn
  - Use the virtual console device n.
- ttySn[,options], ttyUSB0[,options]
  - Use the specified serial port. The options are of the form bbbbpnf, where bbbb is the baud rate, p is parity (n, o, or e), n is number of bits, and f is flow control (r for RTS or omitted). Default is 9600n8.
  - See the file Documentation/serial-console.txt for more information on how to use a serial console. If you wish to have access to the kernel console information and do not have a serial port, see the netconsole command-line option.
- uart,io,addr[,options], uart,mmio,addr[,options]
  - Start an early, polled-mode console on the 8250/16550 UART at the specified I/O port or MMIO address, switching to the specified ttyS device later. The options are the same as for ttyS shown earlier.

实例

- quiet [KNL] Disable most log messages
- splash Causes the splash screen to be shown
- vt.handoff=7 Causes the kernel to maintain the current contents of video memory on virtual terminal 7, which is a new "transparent" VT type

```
ge@gewubox:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.12.2 root=UUID=eb5d8aee-6a0f-4257-b5b8-8d6731ba6764
ro quiet splash vt.handoff=7
```

模块参数

(kernel command line)

- usbcore.blinkenlights=1

(modprobe command line)

- modprobe usbcore blinkenlights=1

观察模块参数

```
ge@gewubox:/boot/grub$ modinfo -p video
brightness_switch_enabled: (bool)
allow_duplicates: (bool)
use_bios_initial_backlight: (bool)
```

说明

- 下划线和减号等价
  - log_buf_len=1M print-fatal-signals=1
  - 等价于
  - log-buf-len=1M print_fatal_signals=1
- 如果内容有空格可以用双引号包围
  - param="spaces in here"
- --后的内容会传递给init

55

通过grub配置

grub.cfg grub

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=5
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
#GRUB_CMDLINE_LINUX="text"
```

- /etc/default/grub
- GRUB_CMDLINE_LINUX_DEFAULT
- #sudo grub-update

56


版本简史
源码目录
架构
内核参数
文档和工具链

57


Documentation
众里寻他千百度，那人却在灯火阑珊处
最值得细读的内核资料
内容很丰富
16.8MB(2.2), 20.6MB(4.4)
超越了windows的地方
ABI
accounting
acpi
aoe
arm
arm64
auxdisplay
backlight
blackfin
block
blockdev
bus-devices
cdrom
cgroups
cma
connector
console
cpu-freq
cpuidle
cris
crypto
development-process
device-mapper
devicetree
dmaengine
DocBook
driver-model
dvb
early-userspace
EDID
extcon
fault-injection
fb
features
filesystems
firmware_class
fmc
fpga
frv
gpio
hid
hwmon
i2c
ia64
ide
infiniband
input
ioctl
isdn
ja_JP
kbuild
kdump
ko_KR
laptops
leds
locking
mtd
memory-devices
metag
mic
mips
misc-devices
mmc
mn10300
mtd
namespaces
netlabel
networking
nfc
nios2
nvdimm
nvmem
parisc
PCI
pcmcia
phy
platform
power
powerpc
pps
prctl
pti
ptp
rapidio
RCU
s390
scheduler
scsi
security
serial
sh
sound
spi
sysctl
target
thermal
timers
tpm
trace
usb
vDSO
video4linux
virtual
vm
w1
watchdog
wimax
x86
xtensa
zh_CN
00-INDEX
adding-syscalls.txt
applying-patches.txt
assoc_array.txt
atomic_ops.txt
bad_memory.txt
basic_profiling.txt
bcache.txt
binfmt_misc.txt
braille-console.txt
bt8xxgpio.txt
btmrvl.txt
BUG-HUNTING
bus-virt-phys-mapping.txt
cachetlb.txt
Changes
circular-buffers.txt
clk.txt
coccinelle.txt
CodeOfConflict
CodingStyle
cpu-hotplug.txt
cpu-load.txt
cputopology.txt
crc32.txt
dcdbas.txt
debugging-modules.txt
debugging-via-ohci1394.txt
dell_rbu.txt
devices.txt
digsig.txt
DMA-API.txt
DMA-API-HOWTO.txt
DMA-attributes.txt
dma-buf-sharing.txt
DMA-ISA-LPC.txt
dontdiff
dynamic-debug-howto.txt
edac.txt
efi-stub.txt
eisa.txt
email-clients.txt
flexible-arrays.txt
futex-requeue-pi.txt
gcov.txt
gdb-kernel-debugging.txt
highuid.txt
HOWTO
hsi.txt
hw_random.txt

58

00-index

This is a brief list of all the files in ./linux/Documentation and what they contain. If you add a documentation file, please list it here inalphabetical order as well, or risk being hunted down like a rabid dog.Please keep the descriptions small enough to fit on one line.
Thanks -- Paul G.

Lister - [c:\bench\linux-4.4.14\Documentation\00-INDEX]

File Edit Options Help 10 %

```
This is a brief list of all the files in ./linux/Documentation and what
they contain. If you add a documentation file, please list it here in
alphabetical order as well, or risk being hunted down like a rabid dog.
Please keep the descriptions small enough to fit on one line.
                                                        Thanks -- Paul G.

Following translations are available on the WWW:

   - Japanese, maintained by the JF Project (jf@listserv.linux.or.jp), at
     http://linuxjf.sourceforge.jp/

00-INDEX
        - this file.
ABI/
        - info on kernel <-> userspace ABI and relative interface stability.

BUG-HUNTING
        - brute force method of doing binary search of patches to find bug.
Changes
        - list of changes that break older software packages.
CodingStyle
        - how the maintainers expect the C code in the kernel to look.
DMA-API.txt
        - DMA API, pci_ API & extensions for non-consistent memory machines.
DMA-API-HOWTO.txt
        - Dynamic DMA mapping Guide
DMA-ISA-LPC.txt
        - How to do DMA with ISA (and LPC) devices.
DMA-attributes.txt
        - listing of the various possible attributes a DMA region can have
DocBook/
        - directory with DocBook templates etc. for kernel documentation.
EDID/
        - directory with info on customizing EDID for broken gfx/displays.
HOWTO
        - the process and procedures of how to do Linux kernel development.
IPMI.txt
        - info on Linux Intelligent Platform Management Interface (IPMI) Drive
IRQ-affinity.txt
        - how to select which CPU(s) handle which interrupt events on SMP.
IRQ-domain.txt
        - info on interrupt numbering and setting up IRQ domains.
IRQ.txt
        - description of what an IRQ is.
Intel-IOMMU.txt
        - basic info on the Intel IOMMU virtualization support.
```

59

更多书籍或者在线资源

Lister - [D:\bench\linux-3.16.3\Documentation\kernel-docs.txt]

File Edit Options Help 2 %

```
    Index of Documentation for People Interested in Writing and/or

                    Understanding the Linux Kernel.

          Juan-Mariano de Goyeneche <jmseyas@dit.upm.es>

/*
 * The latest version of this document may be found at:
 *   http://www.dit.upm.es/~jmseyas/linux/kernel/hackers-docs.html
 */

   The need for a document like this one became apparent in the
   linux-kernel mailing list as the same questions, asking for pointers
   to information, appeared again and again.

   Fortunately, as more and more people get to GNU/Linux, more and more
   get interested in the Kernel. But reading the sources is not always
   enough. It is easy to understand the code, but miss the concepts, the
   philosophy and design decisions behind this code.
```

60

c:\bench\linux-4.4.14\Documentation\Abi\stable*.*

↑Name	Ext	Size	Date	Attr
[..]		<DIR>	2016/06/25 01:18	----
firewire-cdev		4,140	2016/06/25 01:18	-a--
o2cb		457	2016/06/25 01:18	-a--
syscalls		442	2016/06/25 01:18	-a--
sysfs-acpi-pmprofile		675	2016/06/25 01:18	-a--
sysfs-bus-firewire		4,285	2016/06/25 01:18	-a--
sysfs-bus-usb		5,051	2016/06/25 01:18	-a--
sysfs-bus-vmbus		969	2016/06/25 01:18	-a--
sysfs-bus-w1		524	2016/06/25 01:18	-a--
sysfs-bus-xen-backend		2,532	2016/06/25 01:18	-a--
sysfs-class-backlight		1,914	2016/06/25 01:18	-a--
sysfs-class-rfkill		1,979	2016/06/25 01:18	-a--
sysfs-class-tpm		6,914	2016/06/25 01:18	-a--
sysfs-class-ubi		6,956	2016/06/25 01:18	-a--
sysfs-class-udc		2,448	2016/06/25 01:18	-a--
sysfs-devices		400	2016/06/25 01:18	-a--
sysfs-devices-node		3,010	2016/06/25 01:18	-a--
sysfs-devices-system-cpu		979	2016/06/25 01:18	-a--
sysfs-devices-system-xen_memory		2,656	2016/06/25 01:18	-a--
sysfs-driver-ib_srp		7,796	2016/06/25 01:18	-a--
sysfs-driver-qla2xxx		377	2016/06/25 01:18	-a--
sysfs-driver-usb-usbtmc		2,407	2016/06/25 01:18	-a--
sysfs-driver-w1_ds28e04		648	2016/06/25 01:18	-a--
sysfs-driver-w1_ds28ea00		305	2016/06/25 01:18	-a--
sysfs-firmware-efi-vars		2,441	2016/06/25 01:18	-a--
sysfs-firmware-opal-dump		1,490	2016/06/25 01:18	-a--
sysfs-firmware-opal-elog		2,209	2016/06/25 01:18	-a--
sysfs-module		1,331	2016/06/25 01:18	-a--
sysfs-transport-srp		2,538	2016/06/25 01:18	-a--
thermal-notification		190	2016/06/25 01:18	-a--
vdso		1,329	2016/06/25 01:18	-a--

二进制接口

Frame Buffer

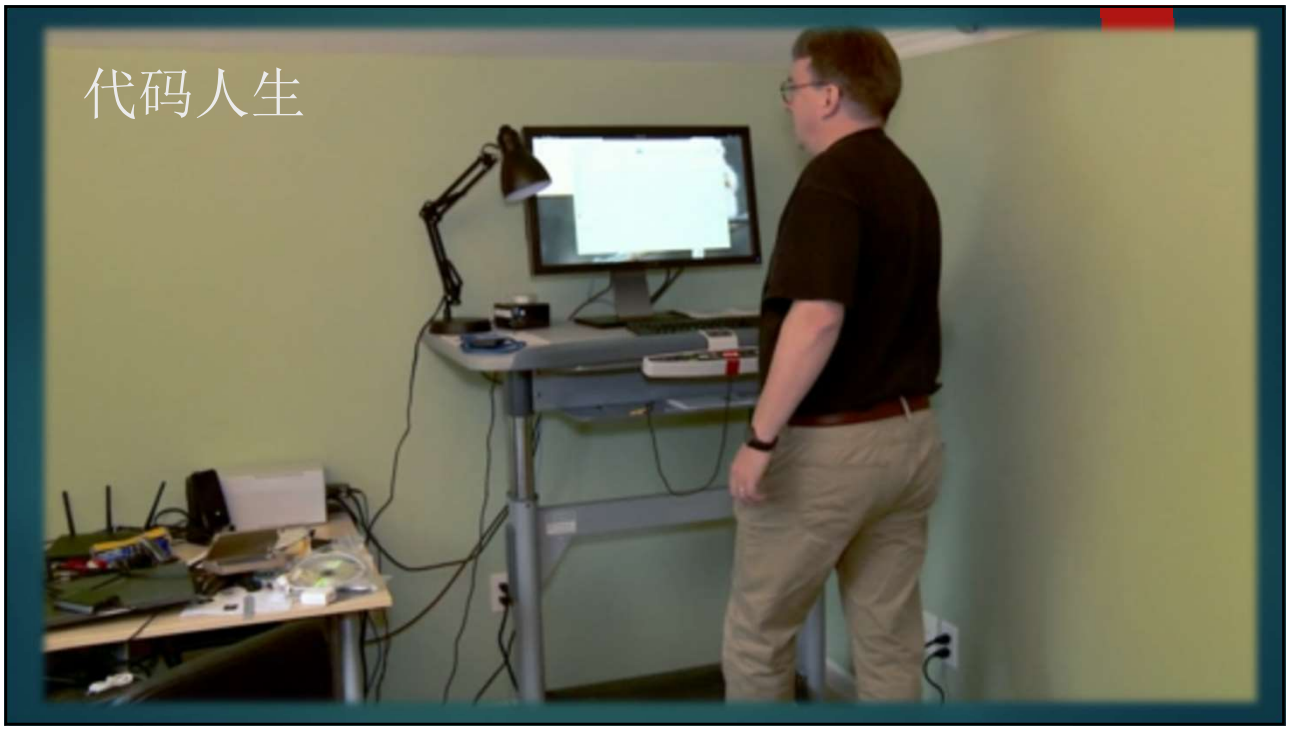
[.] [..] 00-INDEX api.txt
arkfb.txt aty128fb.txt
cirrusfb.txt cmap_xfbdev.txt
deferred_io.txt efifb.txt ep93xx-fb.txt
fbcon.txt
framebuffer.txt gxfb.txt intel810.txt
intelfb.txt internals.txt lxfb.txt
matroxfb.txt metronomefb.txt
modedb.txt pvr2fb.txt pxafb.txt
s3fb.txt
sa1100fb.txt sh7760fb.txt sisfb.txt
sm501.txt sm712fb.txt sstfb.txt
tgafb.txt tridentfb.txt udlfb.txt
uvesafb.txt vesafb.txt viafb.modes
viafb.txt vt8623fb.txt

- 直接写显示缓冲区，古老而且快捷的显示方法

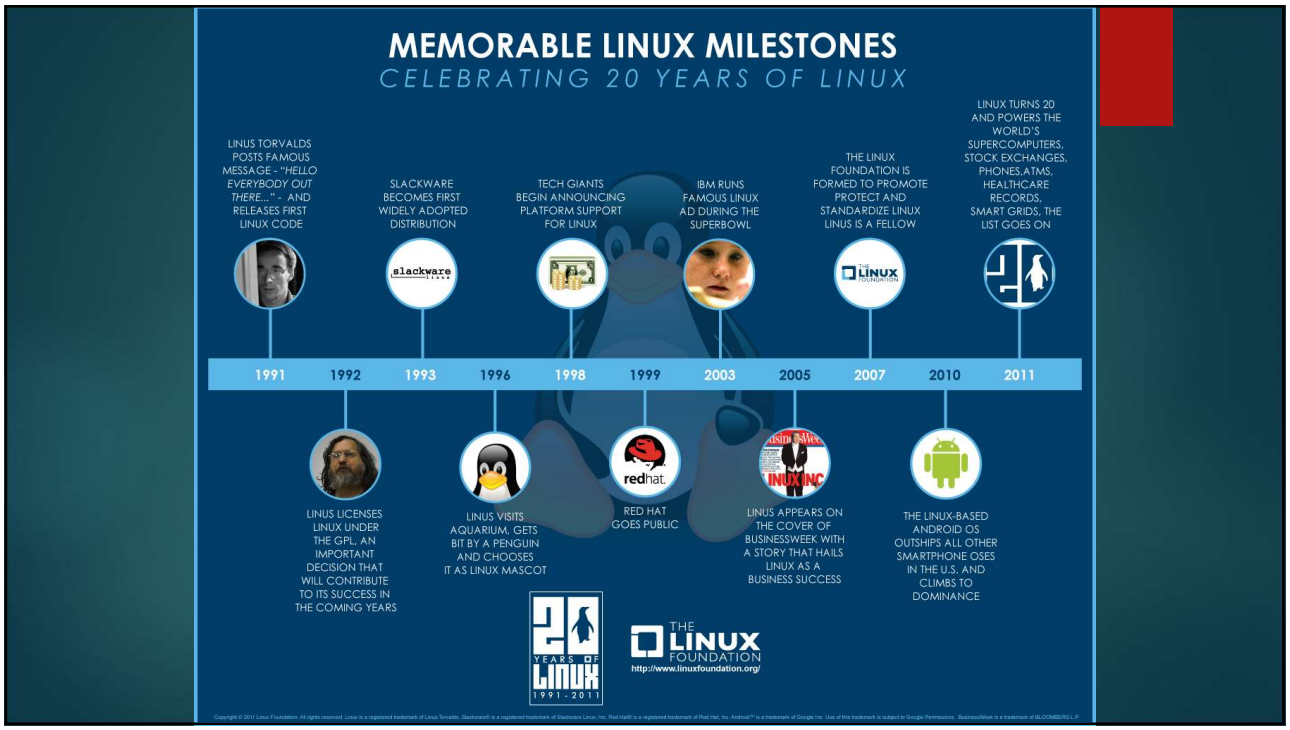
Lister - [c:\bench\linux-4.4.14\Documentation\fb\internals.txt]

File Edit Options Help 58 %

```
This is a first start for some documentation about frame buffer device
internals.

Geert Uytterhoeven <geert@linux-m68k.org>, 21 July 1998
James Simmons <jsimmons@user.sf.net>, Nov 26 2002

--------------------------------------------------------------------------------

            ***  STRUCTURES USED BY THE FRAME BUFFER DEVICE API  ***

The following structures play a role in the game of frame buffer devices. They
are defined in <linux/fb.h>.

1. Outside the kernel (user space)

  - struct fb_fix_screeninfo

    Device independent unchangeable information about a frame buffer device and
    a specific video mode. This can be obtained using the FBIOGET_FSCREENINFO
    ioctl.

  - struct fb_var_screeninfo

    Device independent changeable information about a frame buffer device and a
    specific video mode. This can be obtained using the FBIOGET_VSCREENINFO
    ioctl, and updated with the FBIOPUT_VSCREENINFO ioctl. If you want to pan
    the screen only, you can use the FBIOPAN_DISPLAY ioctl.

  - struct fb_cmap

    Device independent colormap information. You can get and set the colormap
    using the FBIOGETCMAP and FBIOPUTCMAP ioctls.


2. Inside the kernel

  - struct fb_info

    Generic information, API and low level information about a specific frame
    buffer device instance (slot number, board address, ...).

  - struct `par'

    Device dependent information that uniquely defines the video mode for this
    particular piece of hardware.
```


代码人生

63


MEMORABLE LINUX MILESTONES
CELEBRATING 20 YEARS OF LINUX
LINUS TORVALDS POSTS FAMOUS MESSAGE - "HELLO EVERYBODY OUT THERE..." - AND RELEASES FIRST LINUX CODE
SLACKWARE BECOMES FIRST WIDELY ADOPTED DISTRIBUTION
TECH GIANTS BEGIN ANNOUNCING PLATFORM SUPPORT FOR LINUX
IBM RUNS FAMOUS LINUX AD DURING THE SUPERBOWL
THE LINUX FOUNDATION IS FORMED TO PROMOTE PROTECT AND STANDARDIZE LINUX LINUS IS A FELLOW
LINUX TURNS 20 AND POWERS THE WORLD'S SUPERCOMPUTERS, STOCK EXCHANGES, PHONES, ATMS, HEALTHCARE RECORDS, SMART GRIDS, THE LIST GOES ON
1991
1992
1993
1996
1998
1999
2003
2005
2007
2010
2011
LINUS LICENSES LINUX UNDER THE GPL, AN IMPORTANT DECISION THAT WILL CONTRIBUTE TO ITS SUCCESS IN THE COMING YEARS
LINUS VISITS AQUARIUM, GETS BIT BY A PENGUIN AND CHOOSES IT AS LINUX MASCOT
RED HAT GOES PUBLIC
LINUS APPEARS ON THE COVER OF BUSINESSWEEK WITH A STORY THAT HAILS LINUX AS A BUSINESS SUCCESS
THE LINUX-BASED ANDROID OS OUTSHIPS ALL OTHER SMARTPHONE OSES IN THE U.S. AND CLIMBS TO DOMINANCE
20 YEARS OF LINUX 1991-2011
THE LINUX FOUNDATION
http://www.linuxfoundation.org/

64

"Talk is cheap. Show me the code."

Let's CODE!

65

切问而近思

欢迎关注格友公众号

66