

# Kurs administrowania systemem Linux

## Zajęcia nr 15: Zakończenie

Instytut Informatyki Uniwersytetu Wrocławskiego

21 czerwca 2022

# Urządzenia dyskowe

## Zapis na nośniku fizycznym

- magnetyczny na dysku: CMR, SMR (Shingled Magnetic Recording)
- w komórkach NAND flash: Solid State Drive (SSD)

## Kontroler dysku

- odwzorowanie sektorów (sector reallocation, Flash Translation Layer)
- korekcja błędów odczytu (Reed-Solomon, Low Density Parity Check, Turbo Codes, BCH)
- diagnostyka (S.M.A.R.T.)
- obsługa magistrali dysku

## Magistrala dysku

- SATA, NVMe, eMMC, SAS, Fiber Channel

## Host Controller

- Host Controller Interface (SCSI, ATAPI, AHCI, NVMe, OHCI/UHCI/EHCI/xHCI)

## Device Driver

- Tunelowanie np. w USB: SAT (SCSI/ATA Translation), UAS (USB Attached SCSI)

# Organizacja dysku

## Atomowa jednostka zapisu: sektor

- Sektor logiczny: prawie zawsze 512 B (ale są też np. 520 B)
- LBA (Logical Block Addressing), numerowane od 0
- SATA: LBA ma 48 bitów (128 PB); dawniej 22 bity (2 GB), potem 28 bitów (128 GB)
- interfejs jądra używa 64 bitów (8 ZB); dawniej 32 bity (2 TB)
- Sektor fizyczny: Advanced Format (4 kB), flash page (4–8 kB)
- Większe obszary: SMR (strefy zapisu, zwykle 20–40 MB), SSD (erase blocks, zwykle 128–512 stron, tj. 512 kB – 4 MB)

## Wnioski

- System operacyjny przedstawia dysk jako zbiór sektorów numerowanych za pomocą LBA.
- Wyrównanie (alignment) i lokalność przestrzenna ważne dla efektywności zarówno dla dysków magnetycznych, jak i SSD.

# Warstwy abstrakcji oferowane przez jądro Linuksa

## Partycje

- Najstarsza metoda podziału dysku na „mniejsze dyski”.
- Obecnie popularny GPT. Dawniej MBR, BSD, Sun, SGI.

## Partycje logiczne: LVM2

- Można dowolnie „rozcinać”, ale i „sklejać” dyski.
- Jeden z modułów sterownika dm (device mapper).

## Softraid

- Sterownik md (multiple device driver)

## Szyfrowanie dysków

- dm-crypt
- dm-integrity, dm-verity

## Device mapper: ogólny mechanizm transformowania dysków

- Liczne moduły (poza wymienionymi też cache, delay, linear, striped, error, zero, flakey, mirror, multipath, snapshot, zoned i in.).
- Sporo starego kodu, który istnieje poza dm:
  - partycje (acorn, aix, amiga, atari, efi, ibm, karma, ldm, mac, msdos, osf, sgi, sun, sysv68, ultrix)
  - osobny driver md

## Rozwiążanie we FreeBSD: GEOM

- Struktura obiektowa (moduły — klasy)
- producenci → obiekt GEOM → konsumenti
- klasy implementują wszystkie dostępne transformacje dysków

# Utrata danych z dysku

## Przyczyny

- Nieatomowość zapisu na dysk
- Uszkodzenie dysku

## Rozwiązywanie problemu atomowości (metadanych)

- Weryfikacja metadanych (`fsck`)
- Księgowanie (*journalling*)
- *Soft updates*
- COW (*copy on write*)

## Rozwiązywanie problemu uszkodzeń dysków: redundancja

- hardware RAID
- soft RAID

Uwaga: RAID zapewnia HA, ale nie zastępuje backupów!

# Uszkodzenia dysków

## Rodzaje nieprawidłowego działania

- całkowita awaria (dysk nie odpowiada)
- *bad sectors* (dysk zgłasza błędy odczytu)
- *silent errors (bit rotting)*

## Bit rotting

- Teoretycznie niemożliwe, bo dysk używa sum kontrolnych!
- Częsty powód: błędy w oprogramowaniu dysku.

## Jak się zabezpieczyć przed „gniciem bitów”?

- RAID nie pomaga
- Sumy kontrolne jedynym sprawdzonym rozwiązaniem
- ext4 ma sumy kontrolne metadanych; ZFS — wszystkiego
- *scrubbing* — wczesne wykrywanie problemów

# Redundant Array of Independent (or Inexpensive) Disks

## RAID levels

- **RAID 0** — block level stripes:  $n \times$  zapis,  $n \times$  odczyt, redundancja: 0, pojemność: 1
- **RAID 1** — mirrors:  $1 \times$  zapis,  $n \times$  odczyt, redundancja:  $n - 1$ , pojemność:  $1/n$
- **RAID 5** — block-level stripes with distributed parity:  $(n - 1) \times$  zapis,  $n \times$  odczyt, redundancja: 1, pojemność:  $1 - 1/n$
- **RAID 6** — block-level stripes with double distributed parity:  $(n - 2) \times$  zapis,  $n \times$  odczyt, redundancja: 2, pojemność:  $1 - 2/n$

## Stare, niepopularne wersje

- **RAID 2** — bit level stripes with Hamming code: bardzo duże szybkości transmisji, praca synchroniczna (tylko jedna transakcja na raz), kody Hamminga nie mają dużej przewagi nad bitami parzystości
- **RAID 3** — byte level stripes with parity
- **RAID 4** — block level stripes with parity

## Poziomy zagnieżdżone

- **RAID 01** — mirror of stripes
- **RAID 10** — stripes of mirrors
- **RAID 03** — byte-level stripes with parity of stripes
- **RAID 50** — block-level stripes of block-level stripes with distributed parity
- **RAID 60** — block-level stripes of block-level stripes with double distributed parity
- **RAID 100** — stripes of stripes of mirrors

## Poziomy niestandardowe

- wiele niestandardowych typów

# Zalety i wady hardware RAID

## Zalety

- System operacyjny „widzi” macierz jako pojedynczy dysk (nie potrzeba oprogramowania, nie ma kłopotów z bootowaniem itd.).
- Obsługa RAID nie obciąża procesora.
- Podtrzymanie baterijne — kontroler RAID potrafi dokończyć transakcję mimo utraty zewnętrznego zasilania.

## Wady

- Mniej elastyczne, niż rozwiązania software'owe.
- System operacyjny nie jest świadom działania macierzy, nie może planować ułożenia danych na dyskach.

## Softraid: sterownik md i interfejs dm-raid

- Zaimplementowany w jądrze Linuksa.
- Nie potrzebuje wsparcia sprzętowego.
- Oferuje RAID 0,1, 4, 5, 6, 10.
- Konfiguracja: `mdadm(8)`.

## Systemy plików często działają latami

- Każdy system plików jest piękny, gdy jest młody.
- Wiele cykli zapisu i kasowania (total writes przekracza wielokrotnie pojemność dysku) — praca alokatora jest bardzo ważna!
- Systemy plików starzeją się w różny sposób (jak wino lub jak mleko).
- ext4 jest jednym z najlepszych klasycznych systemów plików.

## Awarie

- Bardzo dobry fsck.
- Bardzo dobre księgowanie (JBD2) — na poziomie bloków.
- Od niedawna sumy kontrolne metadanych i scrubbing.
- Snapshotting — obecnie poprzez LVM2.

# RAID a system plików

## Wady stosu protokołów md/dm/lvm2/ext4

- RAID nie wie, które sektory są w użyciu, a które nie.
- W razie niespójności RAID nie wie, która wersja jest prawdziwa (nie radzi sobie z gniciem bitów).
- System plików nie wie, na który z fizycznych dysków dane będą zapisane.

## Rozwiązańe

- Połączyć system plików z *volume managerem*.

# ZFS: The Z File System

- Prace rozpoczęto w 2001 w Sun Microsystems.
- Ogłoszono 14/09/2004, pierwsza wersja 31/10/2005 (Open Solaris).
- Licencja CDDL (open source, ale niekompatybilna z GPL i BSD).
- W 2010 Oracle zamknęło kod.
- Forki open source dalej rozwijane niezależnie.
- Paweł Jakub Dawidek: port do FreeBSD 7 (2008).
- FUSE dla Linuksa (2006), natywny port do Linuksa: ZoL (2008).
- Illumos: fork OpenSolarisa z ZFS.
- Inicjatywa OpenZFS (2013).
- FreeBSD 13 (2021): rebase z Illumosa na OpenZFS. Teraz Linux i FreeBSD mają ten sam upstream.

# Założenia projektowe

- „The Enterprise's computer on Star Trek probably runs ZFS” (Michael W. Lucas).
- Liczniki nie 64-bitowe (16 EB, jak w ext4 i Btrfs), ale 128-bitowe. System plików może mieć  $288 \times 10^{15}$  ZB = 288 biliardów ZB = 256 PiZiB.
- Każdy blok (zarówno danych, jak i metadanych) posiada sumy kontrolne.
- Transakcyjność poprzez COW.
- Zalety COW: tanie snapshoty, klony i mirrory, duża lokalność zapisów.
- Wady COW: mała lokalność odczytów.
- Volume manager pozwala na striping, mirroring i stripes with distributed parity (redundancja od 1 do 3 dysków) — odpowiedniki RAID 0, 1, 5 i ich kombinacje.
- Gnicie bitów: problemy w razie uszkodzenia RAM, dlatego zaleca się ECC.
- Wiele niezależnych systemów plików i urządzeń blokowych w jednej przestrzeni dyskowej.
- Kłoci się z hardware RAID.

## Idea trwałych struktur danych

- Neil Sarnak, Robert E. Tarjan, Planar point location using persistent search trees, CACM 29(7):669–679, July 1986.
- James R. Driscoll, Neil Sarnak, Daniel D. K. Sleator, Robert E. Tarjan, Making data structures persistent. J. Comp. System Sci. 38(1):86-124, February 1989.

## Copy on Write

- COW na drzewach jest bardzo efektywny
- Implementacje: ZFS, Btrfs

# Virtual devices (vdev)

## Rodzaje vdev-ów

- disk
- file
- mirror
- raidz1, raidz2, raidz3
- log (ZIL SLOG)
- cache (Level 2 Adaptive Replacement Cache, L2ARC)

Ponadto:

- spare
- draidz1, draidz2, draidz3; non-default draid (hot spare)
- dedup
- special

Przykład (odpowiednik RAID 10):

```
zpool create mypool mirror da0 da1 mirror da2 da3
```

## Datasets i zvole

- Dataset — system plików umieszczony na pewnym zpoolu.
- Mountpoint: domyślnie `/zpoolname/datasetname/`.
- Można mieć datasets niemontowalne.
- Zvol — urządzenie blokowe umieszczone na pewnym zpoolu.
- Uwaga: nie konfigurować swap-a jako zvola!

## Administrowanie ZFS-em

- ZFS przypomina SystemD: zamiast kompatybilnie dodawać — zastępuje.
- Administrowanie LVM-em zrobione na wzór ZFS.
- Zarządzanie pool-ami: polecenie `zpool`.
- Zarządzanie datasetami: polecenie `zfs`.

# Administrowanie ZFS-em we FreeBSD — zpools

- `geom disk list`
- `camcontrol devlist`; CAM (Common Access Method) — tylko dyski SCSI
- `gpart show`; dokładniej: `gpart list`
- `diskinfo -v <dysk>`
- `zpool create -m none tank da0 da1 da2 da3`
- `zpool list`
- `zpool status tank`
- `zpool destroy tank`
- `zpool create -m none tank mirror da0 da1`
- `zpool add tank mirror da2 da3`
- `zpool online tank da2`
- `zpool scrub tank`

# Administrowanie ZFS-em we FreeBSD — datasety, snapshoty i klony

- `zfs create -o mountpoint=/test1 tank/test1`
- `zfs list`
- `zfs create tank/test1/test2`
- `zfs snapshot tank/set1@snap1`
- `zpool set listsnapshots=on tank`
- `zfs clone tank/set1@snap1 tank/set2`
- `zfs set mountpoint=/nowy tank/stary`
  
- `beadm list`

# Administrowanie ZFS-em we FreeBSD — mirrorry i szyfrowanie

- `zfs send tank/set1@snap1 > snap1`
- `cat snap1 | zfs receive barrel/set1`
- `zfs create -o encryption=on -o keylocation=prompt -o keyformat=passphrase pool/dataset`
- `zfs get encryption pool/dataset`
- `zfs load-key -r pool/dataset; zfs mount pool/dataset`
- albo: `zfs mount -l pool/dataset`
- `zfs unload-key -r pool/dataset`
- `zfs get keystate pool/dataset`

*Richard Stallman:* Free Software is about defending the freedom of users.

*Eric Raymond:* Open Source is a better way of doing software development.

*Linus Torvalds:* I wouldn't mind a free beer.

(Panel session at a conference.)



- Organizacja *non-profit* założona w roku 2000 w celu standaryzacji, wspierania i promocji Linuksa. Zarządza:
- The Linux Kernel Organization założoną w 2002 roku.
- Skupia ponad 1000 firm, prowadzi ponad 150 projektów. Jednym z nich jest jądro Linuksa.

# Dlaczego jądro Linuksa?

Statystyki [git.kernel.org](https://git.kernel.org) z 15 września 2018 (Michael Larabel)

Całkowita liczba commitów	782,487
Całkowita liczba autorów	19,009
Liczba plików	61,725
Liczba wierszy	25,584,633

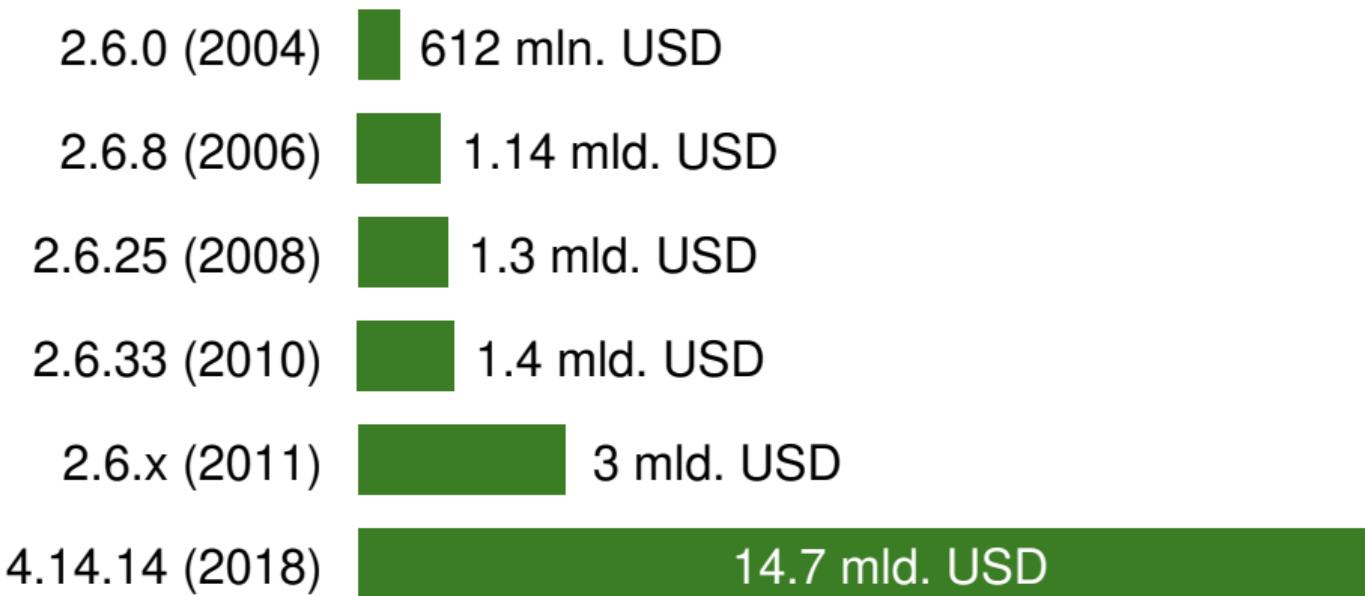
Smartfony, tablety, laptopy i desktopy razem (Gartner 2015)

Android	54.16%
iOS/macOS	12.37%
Windows	11.79%
Inne	21.66%

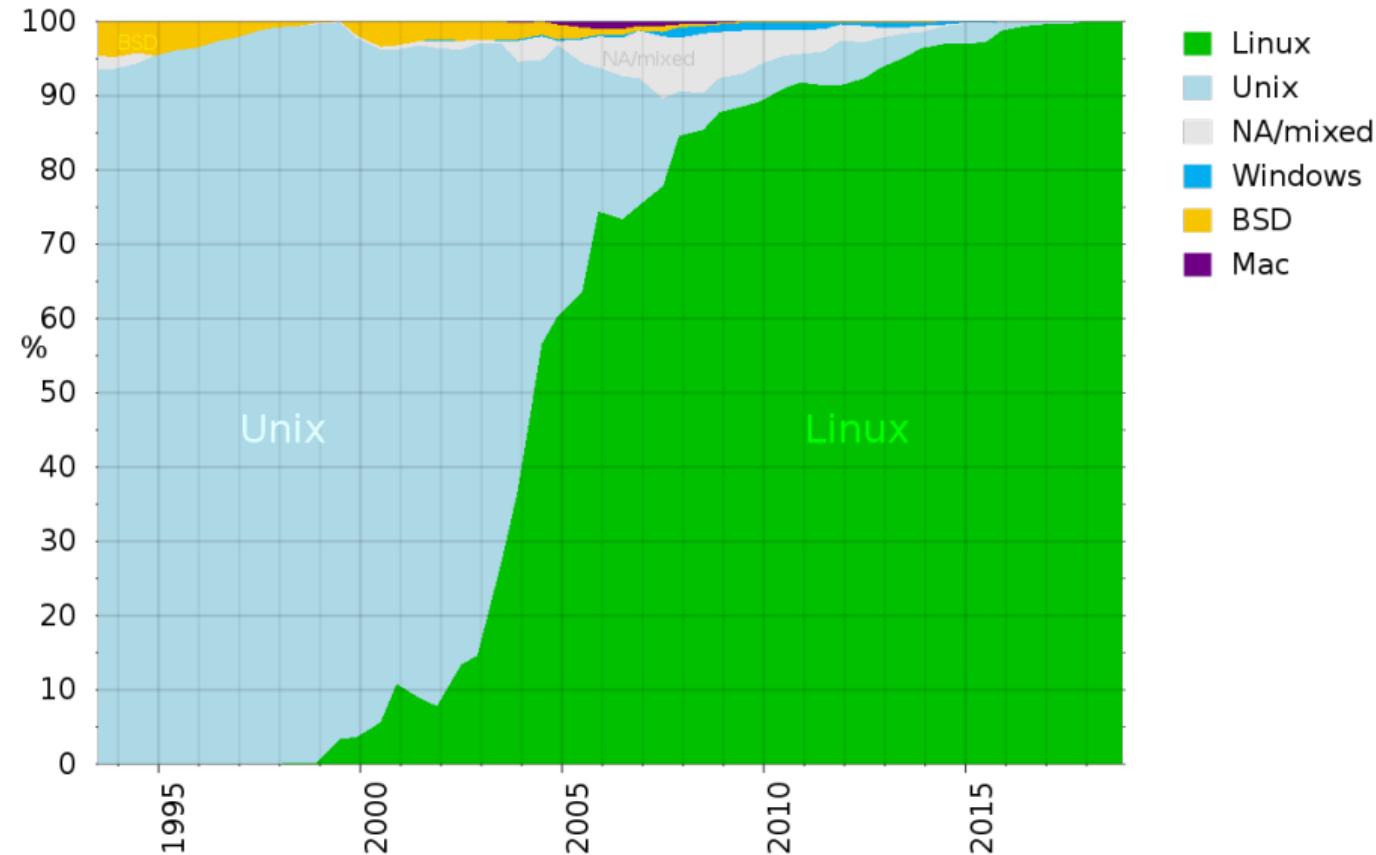
Najwięksi współpracownicy korporacyjni: Intel (12.9%), Red Hat (8%), Linaro (4%), Samsung (3.9%), SUSE (3.2%), and IBM (2.7%).

# Linux kernel redevelopment cost

Gdyby trzeba było zlecić napisanie za pieniądze całego kodu jądra Linuksa od nowa, ile by to kosztowało?



# 500 najszybszych superkomputerów na świecie



# Początki wolnego oprogramowania: Richard Stallman i GNU Project



# GNU Project (36 lat)

- Projekt współtworzenia wolnego oprogramowania przez dużą grupę programistów.
- Założony przez Richarda Stallmana.
- Ogłoszenie: `net.unix-wizards`, `net.usoft`, 27 września 1983 roku.
- Manifest: *Dr Dobbs Journal* 9/1983.
- Cel: stworzyć kompletny system operacyjny będący w całości wolnym oprogramowaniem.
- Nazwa: akronim rekurencyjny *GNU's Not Unix*.
- Slogan: *Free as in freedom*.
- Strona: [www.gnu.org](http://www.gnu.org).
- Jądro systemu: GNU Hurd.
- Rozwiążanie częściowe: GNU/Linux.

# Idea i pierwsze projekty GNU (Stallman)

- W latach '70 rynek oprogramowania stał się na tyle dochodowy, że firmy zaczęły ograniczać dostęp do kodu źródłowego.
- Doświadczenia Stallmana w MIT, swobodny dostęp do kodu, Xerox 9700 (1980).
- Do 1976 roku programy komputerowe w USA były ideami, procedurami, metodami, systemami i procesami i jako takie nie podlegały prawu autorskiemu. US Copyright Act of 1976 to zmienił.
- Projekt GNU był odpowiedzią Stallmana.
- W lutym 1984 roku Stallman zrezygnował z płatnej pracy w MIT (ale zachował biuro) i rozpoczął tworzenie podstawowych pakietów GNU:
  - GNU Emacs
  - GNU Compiler Collection (GCC)
  - GNU Make
  - GNU Debugger
- Początkowo (do 1989 roku) każdy program miał własną licencję.
- W 1989 na podstawie GNU Emacs Public License Stallman stworzył GNU GPL.

# Obecnie dostępne pakiety oprogramowania GNU

3dldf 8sync a2ps acct acm adns alive anubis apl archimedes aris artanis aspell auctex autoconf autoconf-archive autogen automake avl ballandpaddle barcode bash bayonne bazaar bc bfd binutils bison bool bpel2owfn c-graph ccaudio ccd2cue ccide ccrtcp ccscript cflow cgicc chess cim classpath classpathx clisp combine commoncpp complexity config consensus coreutils cpio cppi cssc cursynth dap datamash dc ddd ddrescue dejagnu denemo dia dico diction diffutils dionysus direvent djgpp dominion dr-geo easejs ed edma electric emacs emacs-muse emms enscript eprints epsilon fdisk ferret findutils fisicalab foliot fontopia fontutils freedink freefont freeipmi freetalk fribidi g-golf gama garpd gawk gbehistun gcal gcc gcide gcl gcompris gdb gdbm gengen gengetopt gettext gforth ggradebook ghostscript gift gimp glean gleem global glpk glue gmediaserver gmp gnash gnat gnatsweb gneuralnetwork gnome gnowsys gnu-c-manual gnu-crypto gnu-pw-mgr gnuae gnuastro gnubatch gnubg gnubiff gnubik gnucap gnucash gnucobol gnucomm gnudos gnue gnufm gnugo gnuit gnujdoc gnujump gnukart gnulib gnumach gnumed gnumeric gnump3d gnut gnutet gnupg gnupod gnuprologjava gnuradio gnurobots gnschool gnushogi gnusound gnuspeech gnuspool gnustandards gnustep gnutls gnutrition gnuzilla goptical gorm gpaint gperf gprolog grabcomics greg grep gretl grub gsasl gsegrafix gsl gslip gsrc gss gstick gtk+ gtypist guile guile-cv guile-dbi guile-gnome guile-ncurses guile-opengl guile-rpc guile-sdl guix gurgle gv gvpe gxmessage gzip halifax health hello help2man hp2xx html-info http tunnel hurd hyperbole icecat idutils ignuit indent inetutils inklingreader intfonts jacal java-getopt jel jtw jwhois kawa kopi leg less libc libcdio libdbh liberty-eiffel libextractor libffcall libgcrypt libiconv libidn libjitsi libmatheval libmicrohttpd libredwg librejs libsigsegv libtasn1 libtool libunistring libxmi lightning lilypond lims linux-libre liquidwar6 lispintro lrzsz lsh m4 macchanger mailman mailutils make marst maverik mc mcron mcsim mdk mediagoblin melting mempool metaexchange metahtml metalogic-inference mifluz mig miscfiles mit-scheme moe motti mpc mpfr mpria mtools nana nano nano nano-archimedes ncurses nettle network ocrad octave oleo oo-browser orgadoc osip panorama parallel parted pascal patch paxutils pcb pdf pem pexec pgccfd phantom\_home pies pipo plotutils polyxmass powerguru proxyknife pspp psychosynth pth pyconfigure pythonwebkit qexo quickthreads r radius rcs readline recutils reftex remotecontrol ring rottlog rpge rush sather scm screen sed serveez sharutils shepherd shishi shmm shtool sipwitch slib smalltalk social solfege spacechart speex spell sqltutor src-highlight ssw stalkerfs stow stump superopt swbis sysutils taler talkfilters tar termcap termutils teseq teximpatient texinfo texmacs thales time tramp trans-coord trueprint unifont units unrtf userv uucp vc-dwim vcdimager vera vngen wb wdiff websocket4j webstump wget which womb xaos xboard xhippo xlogmaster xmlat

# Wybrane pakiety oprogramowania GNU

a2ps aspell autoconf autoconf-archive automake bash bc binutils bison combine config  
coreutils cpio dc diffutils emacs enscript fdisk findutils fontutils freefont gawk gcc gcide gdb  
gettext ghostscript gimp gnome gnu-crypto gnumeric gnupg gnutls gprolog grep groff grub  
gsasl gtk+ gv gzip inetutils less libc libgcrypt libiconv libtool linux-libre m4 mailutils make mc  
mtools nano ncurses parallel parted patch pem radius rcs readline screen sed spell sysutils tar  
termcap termutils texinfo time wdiff wget which

Luty 1986

The word “free” in our name does not refer to price; it refers to freedom. First, the freedom to copy a program and redistribute it to your neighbors, so that they can use it as well as you. Second, the freedom to change a program, so that you can control it instead of it controlling you; for this, the source code must be made available to you.

## Współczesna wersja „four freedoms”

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

Dla porównania licencja programu Microsoft Windows zabrania nie tylko udostępniania innym, ale też analizowania (dekompilacji) i modyfikowania kodu programu pod groźbą odpowiedzialności karnej.



- Amerykańska organizacja *non-profit* założona przez Richarda Stallmana 4 października 1985 roku.
- Promuje wolność tworzenia, modyfikowania, rozpowszechniania i analizowania oprogramowania komputerowego.
- Sponsoruje Project GNU.
- Zarządza m. in. licencjami GPL, LGPL i FDL.
- Publikuje listę licencji zgodnych z filozofią *free software*.

# Licencje GNU (33 lata)



**GNU General Public License**  
(ver. 1: 1989, ver. 2: 1991, ver. 3: 2007)



**GNU Lesser (or Library) General Public License**  
(ver. 1: 1989, ver. 2.1: 1991, ver. 3: 2007)



**GNU Free Documentation License**  
(ver. 1: 1999, ver. 1.1: 2000, ver. 1.2: 2002, ver. 1.3: 2008)

Inne: **GNU Affero General Public License, GNU All-Permissive License**

# Inne klasyczne licencje

## Berkeley Software Distribution

- Autor: Regents of the University of California
- Oryginał: 1988, niezgodna z GPL, ale zaakceptowana przez FSF
- *Four-clause version*: 1990, j.w.
- *Three- i Two-clause version*: 1999, zgodne z GPL
- Prostsza i bardziej permisywna niż GPL.

## MIT License

- Najczęściej używana licencja otwartego oprogramowania.
- Podobna do BSD.
- Zawiera *advertising clause*, podobnie jak four-clause BSD.

## Inne ważne

- Apache License 2.0
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

# Linux (31 lat)

From: torvalds@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  
Date: 25 Aug 91 20:57:08 GMT  
Organization: University of Helsinki

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 :-)

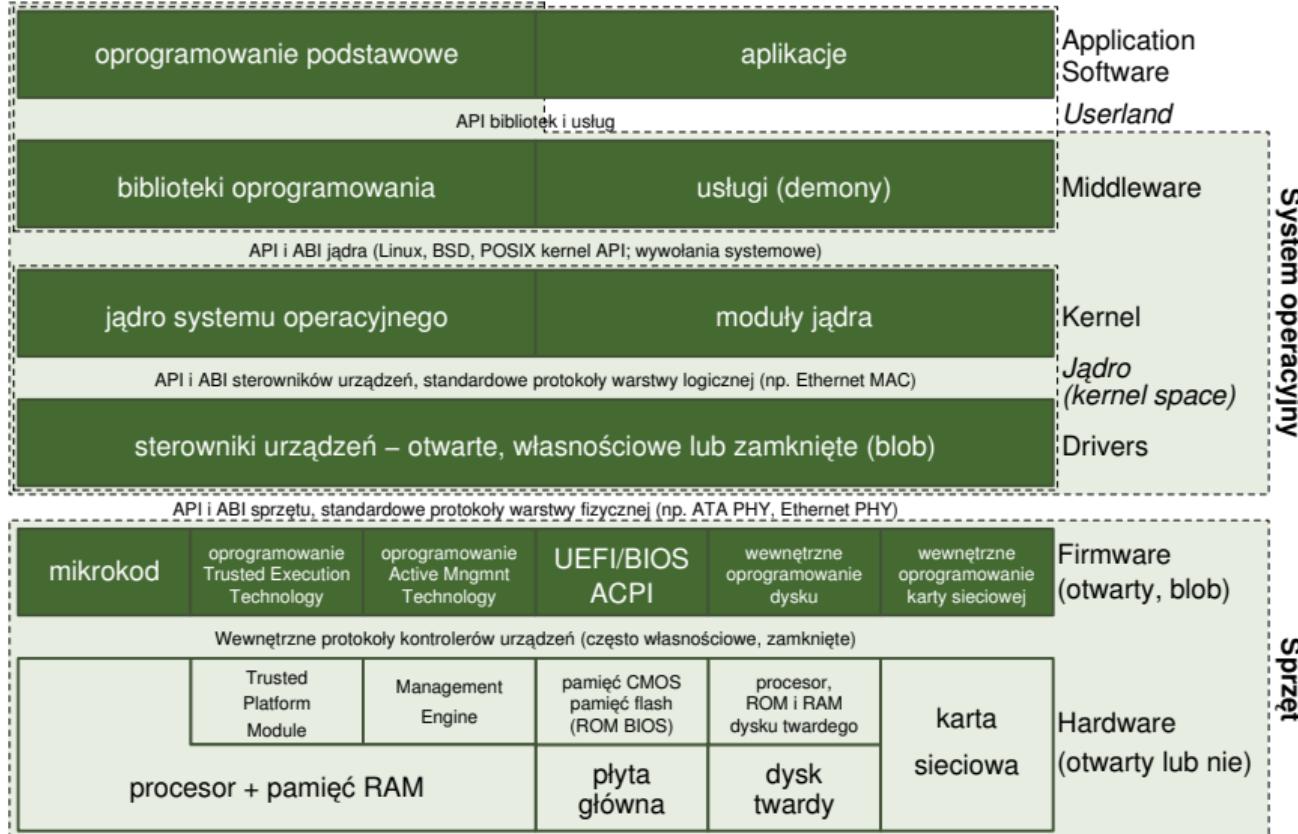
Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

# Licencja Linuksa

- Niewinne ogłoszenie przerodziło się w słynną *Torvalds-Tannenbaum debate* („Linux is obsolete”).
- Wrzesień 1991: wersja 0.01 (10239 wierszy kodu). Cf. 2018: 20 mln. wierszy kodu.
- Od wersji 0.12 na licencji GNU (wcześniej własna licencja Torvalds'a).
- GPL 3 (2007) zawiera klauzule zakazujące m. in. *tivoizacji* kodu. Powstała w odpowiedzi na Digital Millennium Copyright Act.
- Torvalds i wiele innych twórców kodu Linuksa nie zgadza się z nimi. Linux pozostaje przy wersji 2 GPL.

# Hardware i software — ciało i dusza



System operacyjny = jądro + oprogramowanie podstawowe

## Czym się różnią systemy operacyjne?

- Czy przeglądarka WWW należy do systemu operacyjnego?
- KDE  $\neq$  Gnome?
- Debian  $\neq$  Fedora?
- Debian Linux  $\neq$  Debian KFreeBSD?
- Ubuntu = Windows z jądrem Linuksa?
- Jądro jest ukryte — użytkownik odróżnia powłokę.

## Systemy \*niksowe

- \*BSD = jądro + oprogramowanie podstawowe.  
Porty KDE, Gnome itd. jako zewnętrzne programy.  
Podobnie jak MS Windows.
- Linux = różne dystrybucje zawierające jądro Linuksa, KDE, Gnome itd. jako komponenty.
- Czy Android to Linux?

## Dystrybucje oprogramowania

- *Upstream* — oryginalny kod źródłowy autora programu
- Port/pakiet — wersja dostosowana do pracy w określonym środowisku, często przygotowana przez inne osoby, niż autor oryginalnego programu

### Przykład: Mozilla Firefox

- Releases: <https://www.mozilla.org/en-US/firefox/releases>
- Upstream: <https://archive.mozilla.org/pub/firefox/releases>
- Pakiet w Debianie: <https://packages.debian.org/buster/firefox-esr>
- Port we FreeBSD: <https://www.freshports.org/www/firefox>

# Dystrybucje Linuksa



- Kontrowersja: Linux czy GNU/Linux.
- Ubuntu: 8–13% kodu GNU, 9% jądro Linuksa.
- Debian wybrał GNU/Linux.
- Torvalds: GNU Linux jest ok., tak jak Red Hat Linux, czy Slackware Linux.
- Bardzo dużo oprogramowania non-gnu na licencji GPL, np. flex, gnuplot (*sic!*), samo jądro Linuksa.
- Dystrybucje mają swoje własne licencje.
- Problem niezgodności licencji. Przykład: Mozilla Firefox / Debian Iceweasel.

# Debian Free Software Guidelines

Jest częścią *Debian Social Contract*. Opisuje warunki zgodności licencji oprogramowania z DSC i pozwala zdecydować, które oprogramowanie może być włączone (do gałęzi main i contrib) Debiana.

- Free redistribution.
- Inclusion of source code.
- Allowing for modifications and derived works.
- Integrity of the author's source code (as a compromise).
- No discrimination against persons or groups.
- No discrimination against fields of endeavor, like commercial use.
- The license needs to apply to all to whom the program is redistributed.
- License must not be specific to a product.
- License must not restrict other software.

Oprogramowanie o niezgodnej licencji może trafić do non-free, która formalnie nie jest częścią dystrybucji Debiana.

# Open Source Initiative (24 lata)



- Amerykańska organizacja *non-profit* założona przez Erica Raymonda (*The Cathedral and the Bazaar*) i Bruce'a Perensa w lutym 1998 roku.
- Zainspirowani otwarciem kodu Netscape Navigatora.
- Podejście pragmatyczne: nie ideologia („free as in freedom”), tylko pragmatyka — udostępnianie kodu źródłowego na rozsądnej biznesowo licencji.

Stallman

Free software is a political movement; open source is a development model.

# The Open Source Definition (Open Source Initiative)

10 przykazań wzorowanych na DFSG:

- ① Free Redistribution
- ② Source Code
- ③ Derived Works
- ④ Integrity of The Author's Source Code
- ⑤ No Discrimination Against Persons or Groups
- ⑥ No Discrimination Against Fields of Endeavor
- ⑦ Distribution of License
- ⑧ License Must Not Be Specific to a Product
- ⑨ License Must Not Restrict Other Software
- ⑩ License Must Be Technology-Neutral

Publikują własną listę licencji zgodnych z wymaganiami *open source*.

# Ryzyko „bałkanizacji” środowiska wolnego oprogramowania

- Komentarz Torvalds do wersji 3 GPL.
- Wiele niekompatybilnych licencji.
- Dual i multiple licencing.

## Free

- Free of charge.
- Free as in freedom.
- W innych językach niż angielski nie ma dwuznaczności.
- Free/Libre, por. OpenSSL i LibreSSL, OpenOffice i LibreOffice.

## Próby ogólnej klasyfikacji

- FOSS: Free and Open Source
- FLOSS: Free/Libre and Open Source

# Rodzaje licencji

- Public Domain — bez praw (np. SQLite, 7-Zip, SHA-3)
- Non-Protective FOSS License — możliwa wszystko z zachowaniem praw autorskich (np. BSD)
- Protective FOSS License — rozpowszechnianie na tej samej licencji (np. GPL)
- Freeware/Shareware/Freemium (Proprietary License, free) — nie wolno modyfikować, możliwy brak dostępu do kodu źródłowego (np. Irfanview, WinAmp, Google Earth)
- Proprietary License, paid (np. Adobe Photoshop, Microsoft Windows)
- Trade Secret (np. World of Warcraft, server side)

(Według Marka Webbinka)

## Halloween documents

- Tajne dokumenty Microsoftu przedstawiające zagrożenie ze strony Linuksa i opisujące metody walki z FOSS. Wyciek: 31 października 1998.

## Ken Brown: Alexis de Tocqueville Institution (1985–2007)

- *Think tank* zajmujący się badaniem prawa autorskiego i własności intelektualnej.
- Finansowany w latach 1999–2004 roku przez Microsoft.
- Raporty o nieadekwatności oprogramowania FOSS i stratach, jakie przynosi całemu społeczeństwu.
- *Opening the Open Source Debate* (czerwiec 2002): FOSS jest z definicji mniej bezpieczne niż zamknięte oprogramowanie i jest narzędziem terrorystów.
- *Samizdat: And Other Issues Regarding the ‘Source’ of Open Source Code* (2004): Torvalds ukradł kod źródłowy z Miniksa Tanenbauma, bo jeden student nie jest w stanie napisać całego systemu operacyjnego.
- AdTI zakończyło działalność, gdy przewodniczący Gregory Fossedal ogłosił w 2007 roku, że kandyduje na stanowisko prezydenta USA.

# *Closed Source a Open Source: wiara a wiedza*



## Początek jądra Linuksa

- Wiadomość Linusa Torvaldsa na liście comp.os.minix (25/08/1991)
- Linux 0.01 dostępny na ftp.funet.fi (wrzesień 1991)

## Najstarsze dystrybucje

- Softlanding Linux System (SLS, maj 1992).  
Wersja 1.03 z 31/12/1992: 10 dyskietek 1.44MB (łącznie 12MB).
- Slackware (17/07/1993, klon SLS).
- Debian (mail lana Murdocka: 16/08/1993, wersja 0.01: 15/09/1993).
- S.u.S.E Linux (29/03/1994, klon Slackware).
- Red Hat Linux (13/05/1995), Fedora (2003), CentOS (2004).
- Mandrake Linux (23/07/1998), Mandriva (2005), Mageia (2010).



- System pakietów: DEB. Zarządcą pakietów: APT.
- *Community driven*, nie związany z żadnym biznesem.
- Jedna z największych dystrybucji.

## Popularne pochodne



# Debian Census 2019

AIMS Desktop Abalar AlienVault-OSSIM Apertis Aptosid ArcheOS ArchivistaBox Armbian  
AstraLinux BCCD BOSSlinux Bayanihan BlackWeb BlankOn Bloss BunsenLabs Canaima  
Clonezilla CoreBiz Crunchbang CumulusLinux CyborgLinux DebOps Debathena Deepin Devuan  
Donau DoudouLinux Elive Emdebian EmmabuntusDE Endless Epidemic-Linux Finnix  
Freedombone Freeduc GNUSTEP GauSSlan GreenboneOS Grml Hamara Linux HandyLinux  
Huayra Inquisitor Kali Knoppix LMDE Lernstick LiMux Lihuen LinEx LinuxAdvanced Lotos  
Maemo Matriux Meilix MentorEmbeddedLinuxOmniOS MetamorphoseLinux NEMS Linux  
Netrunner OLPC OSMC OpenNetworkLinux Open Secure-K OS Ordissimo PakOS Pardus  
ParrotSecurity Parsix PrimTux ProgressLinux Proxmox PureOS Purism Q4OS Qluster Qubes  
Raspbian Rescatux SELKS SPACEflight SalentOS Septor SerbianLinux SolusOS SolydXK  
SparkyLinux SprezzOS SteamOS StormOS Symbiosis Tails Tanglu ThengOS ToriOS Tucunare  
TurnKeyLinux Ubuntu UltimediaOS UniventionCorporateServer Vanillux Volumio VoyageLinux  
VyOS Vyatta Wazo Webconverger Whonix YunoHost ZevenOS-Neptune Zevenet ev3dev  
gNewSense hLinux rtros semplice siduction ttos ubilinux xanadu



- System pakietów: RPM. Zarządcą pakietów: rpm, yum, dnf.
- Ostatnia wersja: Red Hat Linux 9 (31/03/2003).
- Pochodne: Mandrake Linux (23/06/1998 – 04/2005, francuski remake RHL). Potem Mandriva (do 2011) i Mageia.

## Obecnie

- Red Hat Enterprise Linux (RHEL, produkt komercyjny).
- Fedora (6/11/2003–, *community driven*, wspierany przez RH).
- CentOS (14/05/2004–, *community driven*, obecnie wspierany przez RH).



# Bezpieczeństwo



**QUBES OS**

A REASONABLY SECURE OPERATING SYSTEM

- Bazuje na Fedorze i Xen-ie.
- Rozwijany przez Invisible Things Lab (m. in. Joanna Rutkowska).
- First release: 3 września 2012.
- Bezpieczeństwo przez izolację.

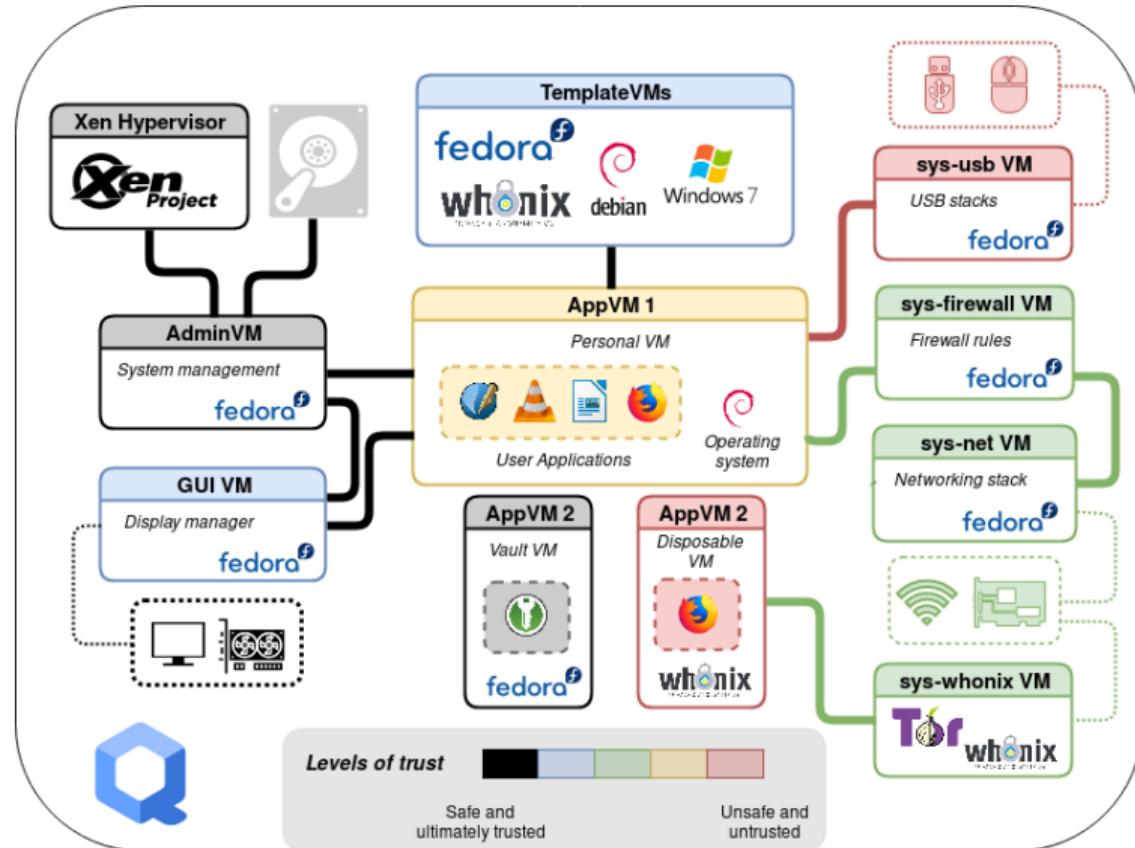


- Bazuje na Debianie.
- Używa Tora do anonimizacji.
- Live CD.



- Bazuje na Debianie.
- Używa Tora do anonimizacji.
- Może działać jako maszyna wirtualna w QubesOS-ie.

# Qubes OS





- Bazuje na Debianie.
- Ogromna liczba narzędzi do pentestingu od razu do użytku.
- Bardzo przydatna dla początkujących hackerów.



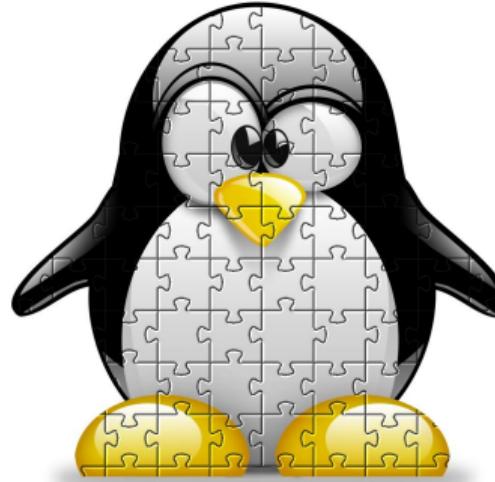
- First release: 11 marca 2002.
- Dla świadomych użytkowników.
- Świadoma instalacja poprzez *bootstrapowanie*. Brak instalatora.
- Duża, dobrze zintegrowana i pomocna społeczność.
- Świetna dokumentacja, wiki i listy dyskusyjne.



# gentoo linux™

- First release: 26 lipca 2000
- Dla świadomych i doświadczonych użytkowników.
- Instalacja poprzez komplikowanie kodu źródłowego — wymaga mocnej maszyny lub komplikacji skrośnej.
- Nie korzysta z SystemD.
- Niewielka, ale dobrze zintegrowana i pomocna społeczność.
- Świetna dokumentacja, wiki i listy dyskusyjne.

# Linux from scratch



- [www.linuxfromscratch.org](http://www.linuxfromscratch.org)
- To nie jest dystrybucja!
- Podręcznik napisany przez Gerarda Beekmansa i rozwijany przez Bruce'a Dubbsa.
- Opis krok po kroku, jak zbudować kompletną instalację Linuksa ze źródeł.
- Kolejne tomy: *Beyond Linux From Scratch*, *Cross Linux From Scratch*, *Hardened Linux From Scratch*, *Automated Linux From Scratch*.



- Bardzo mała dystrybucja, bazuje na musl zamiast glibc i Busybox zamiast Bash/Dash.
- Używa OpenRC, nie używa SystemD.
- Własny system pakietów apk.
- Szczególnie do wirtualizacji i uruchamiania w chmurach.

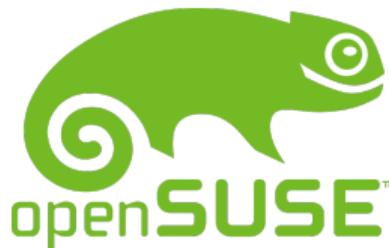
## Inne warte komentarza



- Najstarsza działająca dystrybucja Linuksa
- First release: 17 lipca 1993 (2 miesiące przed Debianem)
- Nie używa SystemD
- Najbardziej BSD-like spośród dystrybucji Linuksa



- First release: grudzień 2002
- Przypomina BSD i Slackware'a
- Nie używa SystemD



- Podobnie jak Fedora wspierana przez korporację (SUSE Linux GmbH) i będąca otwartą wersją dystrybucji komercyjnej
- Dobra dokumentacja i wsparcie
- System pakietów RPM

# Warte wypróbowania

## Stare, ale jare

- Debian
- Fedora
- CentOS
- OpenSUSE

## Nowe

- Gentoo (26/07/2000)
- Arch Linux (11/03/2002)

## Specjalne

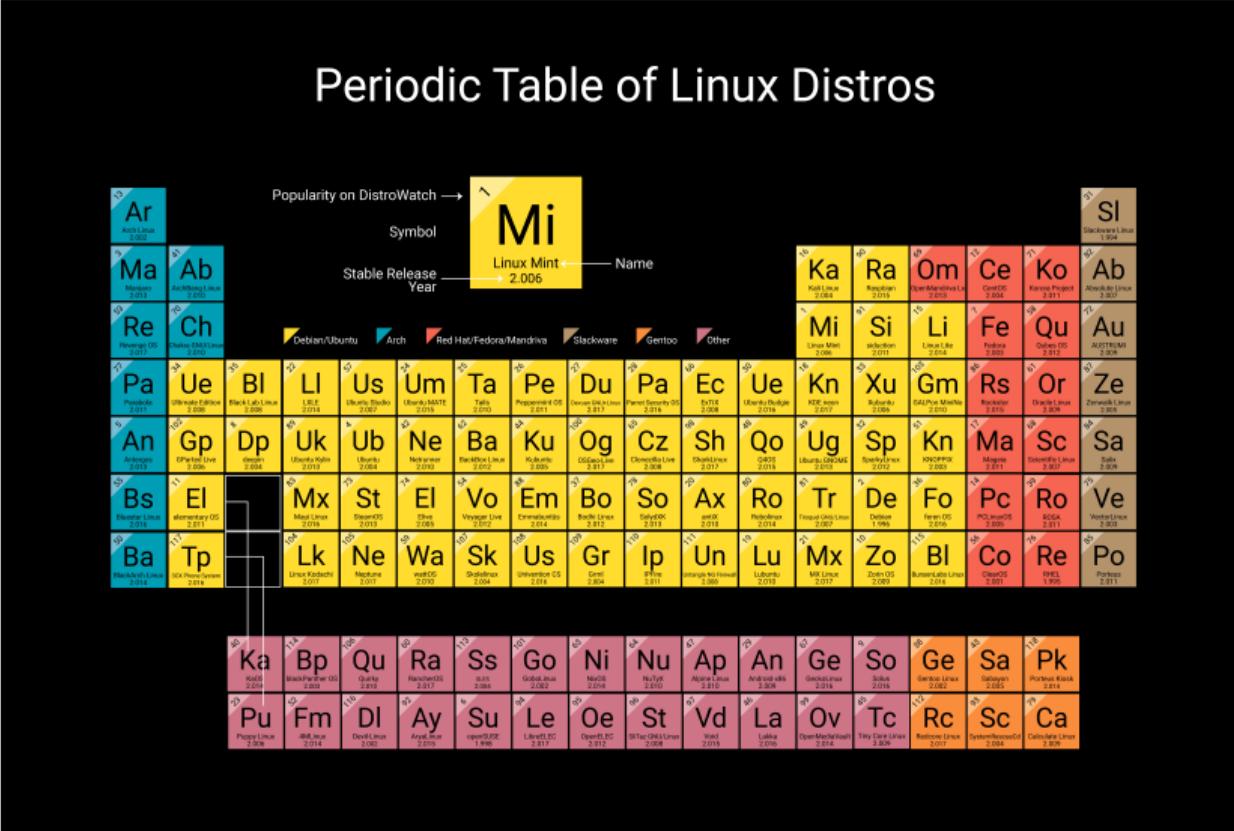
- QubesOS
- Kali
- Alpine Linux
- CRUX
- NixOS

## Dla smakoszy

- Slackware
- Linux From Scratch

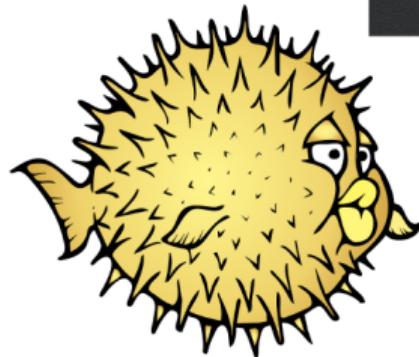
**Najlepsze do nauki administracji** (od najłatwiejszego): Debian, Arch, Gentoo, LFS.

# Periodic Table of Linux Distros (DistroWatch)





FreeBSD



*Open*BSD



# Systemy operacyjne BSD

- *Research Unix* (Bell Labs) był tworzony od 1969 roku.
- University of California at Berkeley kupił licencję od AT&T w 1974 roku.
- UCB wydawał kolejne wersje Berkeley Software Distribution od 1977 roku do 1995 (4.4-Lite2).
- 386BSD (Jolix) — 12 marca 1992, na bazie 4.3BSD Net/2.
- **NetBSD** — 19 kwietnia 1993, na bazie 4.3BSD-Reno, fork 386BSD. *Przenośność*.
- **FreeBSD** — 1 listopada 1993, na bazie 4.3BSD, fork 386BSD. *Skoncentrowany na PC*.
- **OpenBSD** — 1 października 1996, fork NetBSD. *Bezpieczeństwo*.
- **DragonflyBSD** — 12 lipca 2004. *Nowe kierunki*.
- **GhostBSD** — marzec 2010, na bazie FreeBSD. *Łatwy i wygodny*.
- Rozpowszechniane na licencji BSD (dużo prostsza i bardziej permisywna niż GNU).