Continue

You can’t perform that action at this time. Page 2 You can’t perform that action at this time. Page 3 You can’t perform that action at this time. iperf windows builds with GitHub actions Download Latest V3.17.1 V3.17 V3.16 V3.15 V3.14 V3.13 build Build process uses following actions. Build workflow file can be found here build.yaml. iperf Checkout MSYS2 Shell for Windows Automatic Releases LICENSE All upstream assets are Licensed under their respective licenses. iperf’s license is included with the build assets. The iperf series of tools perform active measurements to determine the maximum achievable bandwidth on IP networks. It supports tuning of various parameters related to timing, protocols, and buffers. For each test it reports the measured throughput, loss, and other parameters. This version, sometimes referred to as iperf3, is a redesign of an original version developed at NLANR / DAST. iperf3 is a new implementation from scratch, with the goal of a smaller, simpler code base, and a library version of the functionality that can be used in other programs. iperf3 also incorporates a number of features found in other tools such as nuttcp and netperf, but were missing from the original iperf. These include, for example, a zero-copy mode and optional JSON output. Note that iperf3 is not backwards compatible with the original iperf. Primary development for iperf3 takes place on CentOS Linux, FreeBSD, and macOS. At this time, these are the only officially supported platforms, however there have been some reports of success with OpenBSD, Android, and other Linux distributions. iperf3 is principally developed by ESnet / Lawrence Berkeley National Laboratory. It is released under a three-clause BSD license. iperf2 is no longer being developed by its original maintainers. However, beginning in 2014, another developer began fixing bugs and enhancing functionality, and generating releases of iperf2. Both projects (as of late 2017) are currently being developed actively, but independently. More information can be found in the iperf3 FAQ. Project homepage and documentation hosted on GitHub Pages: Project site (source code repository, issue tracker) hosted on GitHub: Source code downloads: Index Module Index Search Page Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Network performance software tool This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.Find sources: "Iperf" – news · newspapers · books · scholar · JSTOR (March 2020) (Learn how and when to remove this message) iperf2Description of Iperf on TCP port 4662 under linux with an FTTH connection.Original author(s)Mark Gates, Alex WarshavskyDeveloper(s)Robert McMahon, Tim AucklandStable release2.2.1[1] / November 6, 2024 (2024-11-06) Repositorysf.net/p/iperf2/code/Written inCOperating systemCross-platformLicenseBSD licenseWebsitesf.net/projects/iperf2/ iperf3Original author(s)ESnetStable release3.18 / December 13, 2024 (2024-12-13) Repositorygithub.com/esnet/iperfWritten inCOperating systemCross-platformLicenseBSD licenseWebsitesoftware.es.net/iperf/ iperf, Iperf, or iPerf, is a tool for network performance measurement and tuning. It is a cross-platform tool that can produce standardized performance measurements for any network. iperf has client and server functionality, and can create data streams to measure the throughput between the two ends in one or both directions.[2] Typical iperf output contains a time-stamped report of the amount of data transferred and the throughput measured. The data streams can be either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP): UDP: When used for testing UDP capacity, iperf allows the user to specify the datagram size and provides results for the datagram throughput and the packet loss. TCP: When used for testing TCP capacity, iperf measures the throughput of the payload. iperf uses 1024 × 1024 for mebibytes and 1000 × 1000 for megabytes. iperf is open-source software written in C, and it runs on various platforms including Linux, Unix and Windows (either natively or inside Cygwin[3]). The availability of the source code enables the user to scrutinize the measurement methodology. iperf is a compatible reimplementation of the ttcp program that was developed at the National Center for Supercomputing Applications at the University of Illinois by the Distributed Applications Support Team (DAST) of the National Laboratory for Applied Network Research (NLANR),[4] which was shut down on December 31, 2006, on termination of funding by the United States National Science Foundation. iperf3 is a rewrite of iperf from scratch to create a smaller, simpler code base. iperf3 was started in 2009, with the first release in January 2014. iperf3 is not backwards compatible with iperf2. iperf3 also includes a library version which enables other programs to use the provided functionality. iperf3 is single threaded while iperf2 is multi-threaded.[5] Officially iperf3 supports only Linux. Unofficial builds for Windows provided by Vivien Guéant.[6] A user of Neowin, BudMan,[7] provides unofficial Windows builds on his server.[8] Most current Linux distributions have iperf3 in their native package repositories. Unix packages are available from Oracle for Solaris 11.4. Netperf Nuttcp NetPIPE bwping Flowgrind Measuring network throughput Packet generation model ^ "Iperf 2 - Browse Files". SourceForge.net. ^ "iPerf - The TCP, UDP and SCTP network bandwidth measurement tool". Retrieved 21 January 2021. ^ "Cygwin iperf Package". ^ "NLANR/DAST : Iperf - The TCP/UDP Bandwidth Measurement Tool". 2005. Archived from the original on 2008-10-12. ^ "iperf3 FAQ". 2018-07-02. ^ "iPerf - The TCP, UDP and SCTP network bandwidth measurement tool". iperf.fr. Retrieved 2025-02-04. ^ "Iperf 3.18 Windows build". Neowin. 2014-10-28. Retrieved 2025-02-04. ^ "Home". files.budman.pw. Retrieved 2025-02-04. Iperf 2 & Iperf 3 Comparison Table Fenton, Tom (2020-02-20). "Using iPerf to Baseline Network Performance". Schroder, Carla (2007-01-31). "Measure Network Performance with iperf". Retrieved from " How To Use VLC Media Player to Trim Video Clips What Is the $WinREAgent Folder and Can I Delete It? Swear Your Way to Better Search Results How to Get a Dark Start Menu and Taskbar in Windows 10 & 11 Enable, Disable, Manage, Delete or Create a System Restore Point PowerShell and Command Prompt 101 Install All the Microsoft C++ Runtimes at Once with Visual C++ Runtime Installer How to Back up or Restore the Windows Registry How to Fix Error Code 0xc000007b in Windows 10, 8, 7, and Vista How to Fix ERR_SSL_PROTOCOL_ERROR Windows Binaries are available at : macOS : HomeBrew : brew install iperf3 MacPorts : sudo port install iperf3 Android : Building iperf3 for Android : Ubuntu / Debian / Mint launch a terminal and type sudo apt-get install iperf3 Fedora / Red Hat / CentOS / Rocky launch a terminal and type yum install iperf3 FreeBSD launch a terminal and type sudo pkg install benchmarks/iperf3 French forum for iPerf iPerf3 server log script : iperf3tocsv.py (2.5 KiB) by Kirth Gersen Log for iPerf3 : display "date,ip,localport,remoteport,duration,protocol,num_streams,cookie,sent,sent_mbps,rcvd,rcvd_mbps,totalsent,totalreceived" Removing the old version: sudo apt remove iperf3 libiperf0 Install the dependency: sudo apt install libsctp1 Take a recent Ubuntu distribution from source/iperf3 Download iperf3_3.xx-1_amd64.deb and libiperf0_3.xx-1_amd64.deb packages (use amd64 version for a standard version of Ubuntu) Install downloaded packages: sudo dpkg -i libiperf0_3.xx-1_amd64.deb iperf3_3.xx-1_amd64.deb Remove downloaded packages that are now unnecessary: rm libiperf0_3.xx-1_amd64.deb iperf3_3.xx-1_amd64.deb Type of free software license BSD licenses are a family of permissive free software licenses, imposing minimal restrictions on the use and distribution of covered software. This is in contrast to copyleft licenses, which have share-alike requirements. The original BSD license was used for its namesake, the Berkeley Software Distribution (BSD), a Unix-like operating system. The original version has since been revised, and its descendants are referred to as modified BSD licenses. BSD is both a license and a class of license (generally referred to as BSD-like). The modified BSD license (in wide use today) is very similar to the license originally used for the BSD version of Unix. The BSD license is a simple license that merely requires that all code retain the BSD license notice if redistributed in source code format, or reproduce the notice if redistributed in binary format. The BSD license (unlike some other licenses e.g. GPL) does not require that source code be distributed at all. In addition to the original (4-clause) license used for BSD, several derivative licenses have emerged that are also commonly referred to as a "BSD license". Today, the typical BSD license is the 3-clause version, which is revised from the original 4-clause version. In all BSD licenses as following, is the year of the copyright. As published in BSD, is "Regents of the University of California". Prior BSD LicenseAuthorRegents of the University of CaliforniaPublisherPublic domainPublished1988SPDX identifierBSD-4.3TAHOEDebian FSG compatibleYesOSI approvedNoGPL compatibleNoCopyleftNoLinking from code with a different licenceYes Some releases of BSD prior to the adoption of the 4-clause BSD license used a license that is clearly ancestral to the 4-clause BSD license. These releases include some parts of 4.3BSD-Tahoe (1988), about 1000 files,[1] and Net/1 (1989). Although largely replaced by the 4-clause license, this license can be found in 4.3BSD-Reno, Net/2, and 4.4BSD-Alpha. Copyright (c) . All rights reserved. Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the . The name of the may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. BSD LicenseAuthorRegents of the University of CaliforniaPublisherPublic domainPublished1990SPDX identifierBSD-4-Clause(see list for more[2])Debian FSG compatibleYes[3]FSF approvedYes[4]OSI approvedNo[5]GPL compatibleNo[4]CopyleftNo[4]Linking from code with a different licenceYes The original BSD license contained a clause not found in later licenses, known as the "advertising clause". This clause eventually became controversial, as it required authors of all works deriving from a BSD-licensed work to include an acknowledgment of the original source in all advertising material. This was clause number 3 in the original license text:[6] Copyright (c) , All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the . Neither the name of the nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.[6] This clause was objected to on the grounds that as people changed the license to reflect their name or organization it led to escalating advertising requirements when programs were combined in a software distribution: every occurrence of the license with a different name required a separate acknowledgment. In arguing against it, Richard Stallman has stated that he counted 75 such acknowledgments in a 1997 version of NetBSD.[7] In addition, the clause presented a legal problem for those wishing to publish BSD-licensed software which relies upon separate programs using the GNU GPL: the advertising clause is incompatible with the GPL, which does not allow the addition of restrictions beyond those it already imposes; because of this, the GPL's publisher, the Free Software Foundation, recommends developers not use the license, though it states there is no reason not to use software already using it.[4] See also: University of Illinois/NCSA Open Source License New BSD LicenseAuthorRegents of the University of CaliforniaPublisherPublic domainPublished22 July 1999[6]SPDX identifierBSD-3-Clause(see list for more[2])Debian FSG compatibleYes[3]FSF approvedYes[8]OSI approvedYes[5]GPL compatibleYes[8]CopyleftNo[8]Linking from code with a different licenceYes The advertising clause was removed from the license text in the official BSD license on July 22, 1999, by William Hoskins, Director of the Office of Technology Licensing for UC Berkeley.[6][9] [10] On January 31, 2012, UC Berkeley Executive Director of the Office of Intellectual Property and Industry Alliances established that licensees and distributors are no longer required to include the acknowledgement within advertising materials. Accordingly, the advertising clause 3 of the original 4-clause BSD license for any and all software officially licensed under a UC Berkeley version of the BSD license, was deleted in its entirety.[11] Other BSD distributions removed the clause, but many similar clauses remain in BSD-derived code from other sources, and unrelated code using a derived license. While the original license is sometimes referred to as the "BSD-old", the resulting 3-clause version is sometimes referred to by "BSD-new." Other names include new BSD, "revised BSD", "BSD-3", or "3-clause BSD". This version has been vetted as an Open source license by the OSI as "The BSD License".[5] The Free Software Foundation, which refers to the license as the "Modified BSD License", states that it is compatible with the GNU GPL. The FSF encourages users to be specific when referring to the license by name (i.e. not simply referring to it as "a BSD license" or "BSD-style") to avoid confusion with the original BSD license.[8] This version allows unlimited redistribution for any purpose as long as its copyright notices and the license's disclaimers of warranty are maintained. The license also contains a clause restricting use of the names of contributors for endorsement of a derived work without specific permission. Copyright Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.[8] "FreeBSD License" redirects here. For documentation, see FreeBSD Documentation License. See also: ISC license and MIT License FreeBSD LicenseAuthorThe FreeBSD ProjectPublisherThe FreeBSD ProjectPublishedApril 1999 or earlierSPDX identifierBSD-2-Clause(see list for more[2])Debian FSG compatibleYesFSF approvedYes[12]OSI approvedYes[5]GPL compatibleYes[12]CopyleftNo[12]Linking from code with a different licenceYes An even more simplified version has come into use, primarily known for its usage in FreeBSD.[13] It was in use there as early as 29 April 1999[14] and likely well before. The primary difference between it and the New BSD (3-clause) License is that it omits the non-endorsement clause. The FreeBSD version of the license also adds a further disclaimer about views and opinions expressed in the software,[15] though this is not commonly included by other projects. The Free Software Foundation, which refers to the license as the FreeBSD License, states that it is compatible with the GNU GPL. In addition, the FSF encourages users to be specific when referring to the license by name (i.e. not simply referring to it as "a BSD license" or "BSD-style"), as it does with the modified/new BSD license, to avoid confusion with the original BSD license.[12] Copyright (c) , Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.[13] Other projects, such as NetBSD, use a similar 2-clause license.[16] This version has been vetted as an Open source license by the OSI as the "Simplified BSD License."[5] The ISC license without the 'and/or' wording is functionally equivalent, and endorsed by the OpenBSD project as a license template for new contributions.[17] BSD Zero Clause LicenseAuthorRob LandleyPublished2013SPDX identifier0BSDDebian FSG compatibleYesFSF approvedYes[18]OSI approvedYes[19]GPL compatibleYes[18]CopyleftNoLinking from code with a different licenceYes The BSD 0-clause license goes further than the 2-clause license by dropping the requirements to include the copyright notice, license text, or disclaimer in either source or binary forms. Doing so forms a public-domain-equivalent license,[20] the same way as MIT No Attribution License.[citation needed] It is known as "0BSD", "Zero-Clause BSD", or "Free Public License 1.0.0".[21][22] It was created by Rob Landley and first used in Toybox when he was disappointed after using the GNU General Public License in BusyBox.[23] Copyright (C) [year] by [copyright holder] Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted. THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.[22] The SPDX License List contains extra BSD license variations. Examples include:[2] BSD-1-Clause, a license with only the source code retaining clause, used by Berkeley Software Design in the 1990s,[24][25] and later used by the Boost Software License. OSI approved since 2020.[26] BSD-2-Clause-Patent, a variation of BSD-2-Clause with a patent grant. OSI approved since 2017.[27] BSD-3-Clause-No-Nuclear-Warranty, a variation of BSD-3-Clause that adds that a piece of software is not licensed for use in a nuclear facility. Main article: License compatibility The FreeBSD project argues on the advantages of BSD-style licenses for companies and commercial use-cases due to their license compatibility with proprietary licenses and general flexibility, stating that the BSD-style licenses place only "minimal restrictions on future behavior" and are not "legal time-bombs", unlike copyleft licenses.[28] The BSD License allows proprietary use and allows the software released under the license to be incorporated into proprietary products. Works based on the material may be released under a proprietary license as closed source software, allowing usual commercial usages under them. The 3-clause BSD license, like most permissive licenses, is compatible with almost all FOSS licenses (and as well proprietary licenses).[29][30] Two variants of the license, the New BSD License/Modified BSD License (3-clause),[8] and the Simplified BSD License/FreeBSD License (2-clause)[12] have been verified as GPL-compatible free software licenses by the Free Software Foundation, and have been vetted as open source licenses by the Open Source Initiative.[5] The original, 4-clause BSD license has not been accepted as an open source license and, although the original is considered to be a free software license by the FSF, the FSF does not consider it to be compatible with the GPL due to the advertising clause.[4] Over the years I've become convinced that the BSD license is great for code you don't care about. I'll use it myself. If there's a library routine that I just want to say 'hey, this is useful to anybody and I'm not going to maintain this,' I'll put it under the BSD license. -- Linus Torvalds at LinuxCon 2016[31] The BSD license family is one of the oldest and most broadly used license families in the free and open-source software ecosystem, and has been the inspiration for a number of other licenses. Many FOSS software projects use a BSD license, for instance the BSD OS family (FreeBSD etc.), Google's Bionic or Toybox. As of 2015[update] the BSD 3-clause license ranked in popularity number five according to Black Duck Software[32] and sixth according to GitHub data.[33] Free and open-source software portal Comparison of free and open-source software licenses Software using the BSD license (category) ^ Bostic, Keith (15 June 1988). "4.3BSD-tahoe release". Newsgroup: comp.sys.tahoe. Retrieved 5 December 2021. ^ a b c d "SPDX License List". spdx.org. SPDX Working Group. ^ a b "License information". Debian. Retrieved 18 February 2010. ^ a b c d e "Original BSD license". Various Licenses and Comments about Them. Free Software Foundation. Retrieved 2 October 2010. ^ a b c d e f "The BSD License:Licensing". Open Source Initiative. 31 October 2006. Archived from the original on 29 November 2009. Retrieved 6 December 2009. ^ a b c d "To All Licensees, Distributors of Any Version of BSD". University of California, Berkeley. 22 July 1999. Retrieved 15 November 2006. ^ Richard Stallman. "The BSD License Problem". Free Software Foundation. Archived from the original on 12 November 2006. Retrieved 15 November 2006. ^ a b c d e f "Modified BSD license". Various Licenses and Comments about Them. Free Software Foundation. Retrieved 2 October 2010. ^ "Berkeley removes Advertising Clause – Slashdot". bsd.slashdot.org. 2 September 1999. Retrieved 2 September 2021. ^ Comparing the BSD and GPL Licenses on Technology Innovation Management Review by Bruce Montague (on October 2007) ^ Katz, Michael. "Executive Director, Office of Intellectual Property and Industry Research Alliances, University of California, Berkeley" (PDF). Office of Intellectual Property and Industry Research Alliances (IPIRA). University of California, Berkeley. Retrieved 19 November 2024. ^ a b c d e "FreeBSD license". Various Licenses and Comments about Them. Free Software Foundation. Retrieved 2 October 2010. ^ a b "The FreeBSD Copyright". The FreeBSD Project. Archived from the original on 25 November 2009. Retrieved 6 December 2009. ^ "The FreeBSD Copyright (as available at archive.org)". The FreeBSD Foundation. Archived from the original on 29 April 1999. Retrieved 7 January 2017.{{cite web}}: CS1 maint: bot: original URL status unknown (link) ^ "The FreeBSD Copyright". freebsd.org. Retrieved 25 March 2020. ^ "NetBSD Licensing and Redistribution". The NetBSD Foundation. Retrieved 6 December 2009. ^ "OpenBSD Copyright Policy". Retrieved 17 July 2016. ^ a b "Zero-clause license". Various Licenses and Comments about Them. Free Software Foundation. Retrieved 2 October 2010. ^ "[License-review] Please rename "Free Public License-1.0.0" to 0BSD". Retrieved 15 February 2019. ^ "BSD 0-Clause License (0BSD) Explained in Plain English". tldrlegal.com. Retrieved 15 February 2019. ^ "BSD Zero Clause License". spdx.org. Retrieved 19 February 2021. ^ a b "Zero-Clause BSD / Free Public License 1.0.0 (0BSD)". opensource.org. 5 December 2015. Retrieved 19 February 2021. ^ Toybox vs BusyBox - Rob Landley, hobbyist, retrieved 28 April 2023 ^ "BSD 1-Clause License". Software Package Data Exchange (SPDX). 2018. Retrieved 30 May 2022. ^ "Log of /head/include/ifaddrs.h". svnweb.freebsd.org. Retrieved 30 May 2022. ^ "1-clause BSD License – Open Source Initiative". Open Source Initiative. 13 March 2020. Retrieved 26 March 2024. ^ "BSD+Patent – Open Source Initiative". Open Source Initiative. 4 April 2017. Retrieved 26 March 2024. ^ Montague, Bruce (13 November 2013). "Why you should use a BSD style license for your Open Source Project – GPL Advantages and Disadvantages". FreeBSD. Retrieved 28 November 2015. In contrast to the GPL, which is designed to prevent the proprietary commercialization of Open Source code, the BSD license places minimal restrictions on future behavior. This allows BSD code to remain Open Source or become integrated into commercial solutions, as a project's or company's needs change. In other words, the BSD license does not become a legal time-bomb at any point in the development process. In addition, since the BSD license does not come with the legal complexity of the GPL or LGPL licenses, it allows developers and companies to spend their time creating and promoting good code rather than worrying if that code violates licensing. ^ Hanwell, Marcus D. (28 January 2014). "Should I use a permissive license? Copyleft? Or something in the middle?". opensource.com. Retrieved 30 May 2015. Permissive licensing simplifies things One reason the business world, and more and more developers […], favor permissive licenses is in the simplicity of reuse. The license usually only pertains to the source code that is licensed and makes no attempt to infer any conditions upon any other component, and because of this there is no need to define what constitutes a derived work. I have also never seen a license compatibility chart for permissive licenses; it seems that they are all compatible. ^ "Licence Compatibility and Interoperability". Open-Source Software – Develop, share, and reuse open source software for public administrations. joinup.ec.europa.eu. Archived from the original on 17 June 2015. Retrieved 30 May 2015. The licences for distributing free or open source software (FOSS) are divided in two families: permissive and copyleft. Permissive licences (BSD, MIT, X11, Apache, Zope) are generally compatible and interoperable with most other licences, tolerating to merge, combine or improve the covered code and to re-distribute it under many licences (including non-free or "proprietary"). ^ Torvalds at LinuxCon Part III: Permissive Licenses and Org Charts FOSS Force, 2016 ^ "Top 20 licenses". Black Duck Software. 19 November 2015. Archived from the original on 19 July 2016. Retrieved 19 November 2015. 1. MIT license 24%, 2. GNU General Public License (GPL) 2.0 23%, 3. Apache License 16%, 4. GNU General Public License (GPL) 3.0 9%, 5. BSD License 2.0 (3-clause, New or Revised) License 6%, 6. GNU Lesser General Public License (LGPL) 2.1 5%, 7. Artistic License (Perl) 4%, 8. GNU Lesser General Public License (LGPL) 3.0 2%, 9. Microsoft Public License 2%, 10. Eclipse Public License (EPL) 2% ^ Balter, Ben (9 March 2015). "Open source license usage on GitHub.com". github.com. Retrieved 21 November 2015. "1 MIT 44.69%, 2 Other 15.68%, 3 GPLv2 12.96%, 4 Apache 11.19%, 5 GPLv3 8.88%, 6 BSD 3-clause 4.53%, 7 Unlicense 1.87%, 8 BSD 2-clause 1.70%, 9 LGPLv3 1.30%, 10 AGPLv3 1.05% Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable, Marshall Kirk McKusick, in: Open Sources: Voices from the Open Source Revolution, O'Reilly 1999 The Amazing Disappearing BSD License BSD License Definition – by The Linux Information Project (LINFO) Retrieved from " Computer software installed on multiple computing platforms "Cross-platform" redirects here. For the railway station interchange, see cross-platform interchange. For the game term, see cross-platform play. "Multi-platform" redirects here. For the mode of storytelling in television, see multi-platform television. Within computing, cross-platform software (also called multi-platform software, platform-agnostic software, or platform-independent software) is computer software that is designed to work in several computing platforms.[1] Some cross-platform software requires a separate build for each platform, but some can be directly run on any platform without special preparation, being written in an interpreted language or compiled to portable bytecode for which the interpreters or run-time packages are common or standard components of all supported platforms.[2] For example, a cross-platform application may run on Linux, macOS and Microsoft Windows. Cross-platform software may run on many platforms, or as few as two. Some frameworks for cross-platform development are Codename One, ArkUI-X, Kivy, Qt, GTK, Flutter, NativeScript, Xamarin, Apache Cordova, Ionic, and React Native.[3] Main article: Computing platform Platform can refer to the type of processor (CPU) or other hardware on which an operating system (OS) or application runs, the type of OS, or a combination of the two.[4] An example of a common platform is Android which runs on the ARM architecture family. Other well-known platforms are Linux/Unix, macOS and Windows, these are all cross-platform.[4] Applications can be written to depend on the features of a particular platform—either the hardware, OS, or virtual machine (VM) it runs on. For example, the Java platform is a common VM platform which runs on many OSs and hardware types. A hardware platform can refer to an instruction set architecture. For example: ARM or the x86 architecture. These machines can run different operating systems. Smartphones and tablets generally run ARM architecture, these often run Android or iOS and other mobile operating systems. A software platform can be either an operating system (OS) or programming environment, though more commonly it is a combination of both. An exception is Java, which uses an OS-independent virtual machine (VM) to execute Java bytecode. Some software platforms are: Android (ARM64) ChromeOS (ARM32, ARM64, IA-32, x86-64) Common Language Infrastructure (CLI) by Microsoft, implemented in: The legacy .NET Framework that works only on Microsoft Windows. The newer .NET framework (simply called ".NET") that works across Microsoft Windows, macOS, and Linux. Other implementations such as Mono (formerly by Novell and Xamarin[5]) HarmonyOS (ARM64, RISC-V, x86, x64, and LoongArch) iOS ((ARMv8-A)) iPadOS (ARMv8-A) Java Linux (Alpha, ARC, ARM, C-Sky, Hexagon, LoongArch, m68k, Microblaze, MIPS, Nios II, OpenRISC, PA-RISC, PowerPC, RISC-V, s390, SuperH, SPARC, x86, Xtensa) macOS x86, ARM (Apple silicon) Microsoft Windows (IA-32, x86-64, ARM, ARM64) PlayStation 4 (x86), PlayStation 3 (PowerPC) and PlayStation Vita (ARM) Solaris (SPARC, x86) SPARC Unix (many platforms since 1969) Web browsers – mostly compatible with each other, running JavaScript web-apps Xbox Minor, historical AmigaOS (m68k), AmigaOS 4 (PowerPC), AROS (x86, PowerPC, m68k), MorphOS (PowerPC) Atari TOS, MiNT BSD (many platforms; see NetBSDnet,[clarification needed] for example) DOS-type systems on the x86: MS-DOS, PC DOS, DR-DOS, FreeDOS OS/2, eComStation BeOS (PowerPC, x86) Main article: Java (software platform) The Java language is typically compiled to run on a VM that is part of the Java platform. The Java virtual machine (Java VM, JVM) is a CPU implemented in software, which runs all Java code. This enables the same code to run on all systems that implement a JVM. Java software can be executed by a hardware-based Java processor. This is used mostly in embedded systems. Java code running in the JVM has access to OS-related services, like disk input/output (I/O) and network access, if the appropriate privileges are granted. The JVM makes the system calls on behalf of the Java application. This lets users to decide the appropriate protection level, depending on an access-control list (ACL). For example, disk and network access is usually enabled for desktop applications, but not for browser-based applets. The Java Native Interface (JNI) can also be used to access OS-specific functions, with a loss of portability. Currently, Java Standard Edition software can run on Microsoft Windows, macOS, several Unix-like OSs, and several real-time operating systems for embedded devices. For mobile applications, browser plugins are used for Windows and Mac based devices, and Android has built-in support for Java. There are also subsets of Java, such as Java Card or Java Platform, Micro Edition, designed for resource-constrained devices. For software to be considered cross-platform, it must function on more than one computer architecture or OS. Developing such software can be a time-consuming task because different OSs have different application programming interfaces (API). Software written for one OS may not automatically work on all architectures that OS supports. Just because software is written in a popular programming language such as C or C++, it does not mean it will run on all OSs that support that language—or even on different versions of the same OS. Web applications are typically described as cross-platform because, ideally, they are accessible from any web browser: the browser is the platform. Web applications generally employ a client–server model, but vary widely in complexity and functionality. It can be hard to reconcile the desire for features with the need for compatibility. Basic web applications perform all or most processing from a stateless server, and pass the result to the client web browser. All user interaction with the application consists of simple exchanges of data requests and server responses. This type of application was the norm in the early phases of World Wide Web application development. Such applications follow a simple transaction model, identical to that of serving static web pages. Today, they are still relatively common, especially where cross-platform compatibility and simplicity are deemed more critical than advanced functionality. Prominent examples of advanced web applications include the Web interface to Gmail and Google Maps. Such applications routinely depend on additional features found only in the more recent versions of popular web browsers. These features include Ajax, JavaScript, Dynamic HTML, SVG, and other components of rich web applications. Because of the competing interests of compatibility and functionality, numerous design strategies have emerged. Many software systems use a layered architecture where platform-dependent code is restricted to the upper- and lowermost layers. Graceful degradation attempts to provide the same or similar functionality to all users and platforms, while diminishing that functionality to a least common denominator for more limited client browsers. For example, a user attempting to use a limited-feature browser to access Gmail may notice that Gmail switches to basic mode, with reduced functionality but still of use. Some software is maintained in distinct codebases for different (hardware and OS) platforms, with equivalent functionality. This requires more effort to maintain the code, but can be worthwhile where the amount of platform-specific code is high. This strategy relies on having one codebase that may be compiled to multiple platform-specific formats. One technique is conditional compilation. With this technique, code that is common to all platforms is not repeated. Blocks of code that are only relevant to certain platforms are made conditional, so that they are only interpreted or compiled when needed. Another technique is separation of functionality, which disables functionality not supported by browsers or OSs, while still delivering a complete application to the user. (See also: Separation of concerns.) This technique is used in web development where interpreted code (as in scripting languages) can query the platform it is running on to execute different blocks conditionally.[6] Third-party libraries attempt to simplify cross-platform capability by hiding the complexities of client differentiation behind a single, unified API, at the expense of vendor lock-in. Responsive web design (RWD) is a Web design approach aimed at crafting the visual layout of sites to provide an optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices, from mobile phones to desktop computer monitors. Little or no platform-specific code is used with this technique. Cross-platform applications need much more integration testing. Some web browsers prohibit installation of different versions on the same machine. There are several approaches used to target multiple platforms, but all of them result in software that requires substantial manual effort for testing and maintenance.[7] Techniques such as full virtualization are sometimes used as a workaround for this problem. Tools such as the Page Object Model allow cross-platform tests to be scripted so that one test case covers multiple versions of an app. If different versions have similar user interfaces, all can be tested with one test case. Web applications are becoming increasingly popular but many computer users still use traditional application software which does not rely on a client/web-server architecture. The distinction between traditional and web applications is not always clear. Features, installation methods and architectures for web and traditional applications overlap and blur the distinction. Nevertheless, this simplifying distinction is a common and useful generalization. Traditional application software has been distributed as binary files, especially executable files. Executables only support the platform they were built for—which means that a single cross-platform executable could be very bloated with code that never executes on a particular platform. Instead, generally there is a selection of executables, each built for one platform. For software that is distributed as a binary executable, such as that written in C or C++, there must be a software build for each platform, using a toolset that translates—transcompiles—a single codebase into multiple binary executables. For example, Firefox, an open-source web browser, is available on Windows, macOS (both PowerPC and x86 through what Apple Inc. calls a Universal binary), Linux, and BSD on multiple computer architectures. The four platforms (in this case, Windows, macOS, Linux, and BSD) are separate executable distributions, although they come largely from the same source code. In rare cases, executable code built for several platforms is combined into a single executable file called a fat binary. The use of different toolsets may not be enough to build a working executables for different platforms. In this case, programmers must port the source code to the new platform. For example, an application such as Firefox, which already runs on Windows on the x86 family, can be modified and re-built to run on Linux on the x86 (and potentially other architectures) as well. The multiple versions of the code may be stored as separate codebases, or merged into one codebase. An alternative to porting is cross-platform virtualization, where