**PRACTICAL 2**

**OPEN SOURCE LICENSES**

**AIM:**

**Report three different licences and create a breif report about them:**

**BSD**

**HISTORY:**

1. 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 more properly termed modified BSD licenses.
2. 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.
3. The BSD license is a simple license that merely requires that all code licensed under the BSD license be licensed under the BSD license if redistributed in source code format. BSD (unlike some other licenses) does not require that source code be distributed at all.

**IDEA:**

**1**. AT&T, who owned the original Unix implementation, was a publicly regulated monopoly tied up in anti-trust court; it was legally unable to sell a product into the software market. It was, however, able to provide it to academic institutions for the price of media.

**2.**Universities rapidly adopted Unix after an OS conference publicized its availability. It was extremely helpful that Unix ran on the PDP-11, a very affordable 16-bit computer, and was coded in a high-level language that was demonstrably good for systems programming. The DEC PDP-11 had, in effect, an open hardware interface designed to make it easy for customers to write their own OS, which was common. As DEC founder Ken Olsen famously proclaimed, “software comes from heaven when you have good hardware”.

**3.**Unix author Ken Thompson returned to his alma mater, University of California Berkeley (UCB), in 1975 and taught the kernel line-by-line. This ultimately resulted in an evolving system known as BSD (Berkeley Standard Distribution).

**4.** UCB converted Unix to 32-bits, added virtual memory, and implemented the version of the TCP/IP stack upon which the Internet was essentially built. UCB made BSD available for the cost of media, under what became known as “the BSD license”. A customer purchased Unix from AT&T and then ordered a BSD tape from UCB.

**WHAT PROBLEM DOES IT SOLVE:**

A BSD style license is a good choice for long duration research or other projects that need a development environment that:

**a)**has near zero cost

**b)**will evolve over a long period of time

**c)** research funding agencies, such as the NSF, ONR and DARPA, should encourage in the earliest phases of funded research projects, the adoption of BSD style licenses for software, data, results, and open hardware.

**DETAILED LICENSING MODEL:**

A permissive software license, sometimes also called BSD-like or BSD-style license,[2] is a FOSS software license with minimal requirements about how the software can be redistributed. Examples include the BSD licenses .

"The ‘BSD-like’ licenses such as the BSD, MIT, and Apache licenses are extremely permissive, requiring little more than attributing the original portions of the licensed code to the original developers in your own code and/or documentation.".[2]

**WHAT POPULAR SOFTWARE ARE RELEASED UNDER BSD LICENSE:**

**A)BRL-CAD:**

BRL-CAD is a constructive solid geometry (CSG) solid modeling computer-aided design (CAD) system. It includes an interactive geometry editor, ray tracing support for graphics rendering and geometric analysis, computer network distributed framebuffer support, scripting, image-processing and signal-processing tools. The entire package is distributed in source code and binary form.

Although BRL-CAD can be used for a variety of engineering and graphics applications, the package's primary purpose continues to be the support of ballistic and electromagnetic analyses.

**POPULAR NEWS ASSSOCIATED WITH THE BSD LICENSE:**

**Ubuntu Linux and BSD Merge to Form New UbuntuBSD Open Source OS:**

BSD -- the open source, Unix-like operating system kernel that lives in Linux's shadow -- is now coming to the Ubuntu world, thanks to a new open source project called **UbuntuBSD.**

A handful of BSD-based systems have survived into the present: FreeBSD, NetBSD and OpenBSD. But they haven't been nearly as influential, within either the server or desktop markets, as GNU/Linux systems.

Now, a group of programmers wants to combine one of the most popular GNU/Linux distributions, **Canonical's** Ubuntu, with BSD by developing a new operating system called (plainly enough) **UbuntuBSD.**

**POPULARITY:**

In September 2005, the BSD Certification Group, after advertising on a number of mailing lists, surveyed 4,330 BSD users, 3,958 of whom took the survey in English, to assess the relative popularity of the various BSD operating systems. About 77% of respondents used FreeBSD, 33% used OpenBSD, 16% used NetBSD, 2.6% used Dragonfly, and 6.6% used other (potentially non-BSD) systems.

DistroWatch, well known in the Linux community and often used as a rough guide to free operating system popularity, publishes page hits for each of the Linux distributions and other operating systems it covers. As of 4 September 2016, using a data span of the last six months it placed FreeBSD in 21st place with 479 hits per day; TrueOS in 46th place with 244 hits per day; GhostBSD in 47th place with 242 hits, OpenBSD in 75th place with 163 hits per day; MidnightBSD in 118th place with 89 hits per day; and NetBSD in 142nd place with 66 hits per day.

**IMPACT:**

For Coop and Migros, during the years 2010 and 2011 BSD realized an impact study on the implementation of GRASP.

The results of the study were presented and discussed in a round table with the supermarket representatives, suppliers, and stakeholders, the session was moderated by the BSD consultants. Afterwards, BSD developed a monitoring system for the GRASP implementation process, with the objective of measuring the process made amongst suppliers.

**GPL:**

**HISTORY:**

**1.**The GPL was written by Richard Stallman in 1989, for use with programs released as part of the GNU project. The original GPL was based on a unification of similar licenses used for early versions of GNU Emacs (1985),[16] the GNU Debugger and the GNU C Compiler.

**2.** The second version of the license, version 2, was released in 1991. Over the following 15 years, members of the free software community became concerned over problems in the GPLv2 license that could let someone exploit GPL-licensed software in ways contrary to the license's intent.[19] These problems included tivoization (the inclusion of GPL-licensed software in hardware that refuses to run modified versions of its software).

**3.** Version 3 was developed to attempt to address these concerns and was officially released on 29 June 2007.

**IDEA:**

**1.**The text of the GPL is itself copyrighted, and the copyright is held by the Free Software Foundation.

**2.** According to the FSF, "The GPL does not require you to release your modified version, or any part of it. You are free to make modifications and use them privately, without ever releasing them. However, if one releases a GPL-licensed entity to the public, there is an issue regarding linking: namely, whether a proprietary program that uses a GPL library is in violation of the GPL.

**PROBLEMS SOLVED BY GPL LICENSE:**

**1.**GPLv3 improves compatibility with several open source software licenses such as Apache License, version 2.0, and the GNU Affero General Public License, which GPLv2 could not be combined with.

**2.**However, GPLv3 software can only be combined and share code with GPLv2 software if the used GPLv2 license has the optional "or later" clause and the software is upgraded to GPLv3. While the "GPLv2 or any later version" clause is considered by FSF as the most common form of licensing GPLv2 software, for example Toybox developer Rob Landley described it as a lifeboat clause.Software projects licensed with the optional "or later" clause include the GNU Project, while a prominent example without the clause is the Linux kernel.

**DETAILED LICENSING MODEL:**

**1.**The GPL is one of the most exciting, innovative capitalist tools ever created. The GPL breaks down walls between vendors and customers while enabling strong competitive differentiation.

The GPL takes a hardheaded look at software development (and human nature) and works to maximize choice, control and a free market.

**2.**The GPL makes co-creators of vendors and buyers, lessening the sometimes-adversarial relationship between the two.Software under the GPL may be run for all purposes, including commercial purposes and even as a tool for creating proprietary software, for example when using GPL-licensed compilers. Users or companies who distribute GPL-licensed works (e.g. software), may charge a fee for copies or give them free of charge. This distinguishes the GPL from shareware software licenses that allow copying for personal use but prohibit commercial distribution, or proprietary licenses where copying is prohibited by copyright law. The FSF argues that freedom-respecting free software should also not restrict commercial use and distribution (including redistribution): the GPL explicitly states that GPL works may be sold at any price.

**SOFTWARE RELEASED UNDER GPL LICENSE:**

MYSQL-GPL (version 2) or proprietary

WORDPRESS- is released under the GPLv2 (or later) license

GNOME-released under gpl

**POPULAR NEWS REGARDING GPL:**

The Free Software Foundation (FSF)'s Licensing & Compliance Lab has been tracking the public discussion of licensing issues surrounding ZFS and the kernel Linux with interest. We haven't made any statement about it, but we plan to soon.

**POPULARITY:**

As has historically been the case, the free software, copyleft GPLv2 is the most popular license choice according to Black Duck.Popular as the GPL remains, however, it no longer enjoys that kind of advantage. If we group both versions (2 and 3) of the GPL together, the GPL is in use within 37% of the Black Duck surveyed projects. The three primary permissive license choices (Apache/BSD/MIT), on the other hand, collectively are employed by 42%. They represent, in fact, three of the five most popular licenses in use today.

**IMPACT:**

First, many very common UNIX applications, such as GNU Emacs, have been released under the GPL, and are used by countless numbers of users every day.

Second, the open source software movement has taken several ideas promoted by the GPL and modified them slightly. The most important is the idea that software licensing should include access to source code. As we move into a more complex era of computing, this issue becomes important for multiple reasons:

Stability and longevity. If your company invests in a proprietary software package for a mission-critical task, and the company that sold you the software subsequently goes out of business, what do you do? It might mean scrapping a large-scale deployment, which can be very costly. However, with source code, you could have internal company developers charged with maintaining the application, fixing bugs, and even developing new features.

Security. By being able to review the source code, you can be certain that a mission-critical application is secure and doesn't contain any backdoors or other potentially devastating security flaws.

**APACHE**

**HISTORY:**

**VERSION 1.1**

The **Apache License 1.1** was approved by the ASF in 2000: *The primary change from the 1.0 license is in the 'advertising clause' (section 3 of the 1.0 license); derived products are no longer required to include attribution in their advertising materials, but only in their documentation.*

**VERSION 2.0**

The ASF adopted the **Apache License 2.0** in January 2004. The stated goals of the license included *making the license easier for non-ASF projects to use, improving* [*compatibility*](https://en.wikipedia.org/wiki/License_compatibility) *with* [*GPL*](https://en.wikipedia.org/wiki/GNU_General_Public_License)*-based software, allowing the license to be included by reference instead of listed in every file, clarifying the license on contributions, and requiring a patent license on contributions that necessarily infringe a contributor's own patents*.

**IDEA:**

The **Apache License, Version 2.0** (**ALv2**) is a [permissive](https://en.wikipedia.org/wiki/Permissive_free_software_licence) [free software license](https://en.wikipedia.org/wiki/Free_software_license) written by the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF).[[5]](https://en.wikipedia.org/wiki/Apache_License#cite_note-nmr-permissive-5) The Apache License requires preservation of the [copyright](https://en.wikipedia.org/wiki/Copyright) notice and [disclaimer](https://en.wikipedia.org/wiki/Disclaimer). Like other [free software licenses](https://en.wikipedia.org/wiki/Free_software_license), the license allows the user of the software the freedom to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software, under the terms of the license, without concern for [royalties](https://en.wikipedia.org/wiki/Royalties). This makes ALv2 a [FRAND-RF](https://en.wikipedia.org/wiki/Reasonable_and_non-discriminatory_licensing) license. The ASF and its projects release the software they produce under the Apache License and many non-ASF projects are also using the ALv2.

**PROBLEMS SOLVED BY APACHE LICENSE:**

1.**Reusable**   
The Apache- and ASF project-specific stuff should be separated out of the license proper so that both ASF and non-ASF projects can use the license terms unaltered.*Solved in 2.0 by moving project-specific stuff into the NOTICE file.*

2.**Patentprotection**  
It would be nice to have some language in the license that protected us and our users from patent-infringement suits, at the very least from contributors if not in more general ways. *Solved in 2.0.*

3.**Applicable to documentation**

Make the license clearly applicable to both software and documentation. Solved in 2.0.

4.**Trademarks**

Avoid listing a specific set of trade marks, trade names, and service marks within the license, since that caused problems with project-specific licenses and apparent GPL-compatibility. Likewise, provide instruction on how and when Apache marks may or may not be used, since that is our most common source of licensing question. Partly solved in 2.0 by excluding any permission to use trademarks within the license itself and moving related information into the NOTICE file.

**DETAILED LICENSING MODEL:**

1.A **permissive software license** is a [free software](https://en.wikipedia.org/wiki/Free_software) [software licence](https://en.wikipedia.org/wiki/Software_licence) with minimal requirements about how the software can be redistributed. Examples include the [MIT Licence](https://en.wikipedia.org/wiki/MIT_Licence), [BSD licences](https://en.wikipedia.org/wiki/BSD_licences), [Apple Public Source License](https://en.wikipedia.org/wiki/Apple_Public_Source_License) and the [Apache licence](https://en.wikipedia.org/wiki/Apache_licence).

2. Apache licenses are extremely permissive, requiring little more than attributing the original portions of the licensed code to the original developers in your own code and/or documentation.

**SOFTWARES RELEASED UNDER APACHE LICENSE:**

Linux

Apache HTTP Server

Open Office

OpenStack

Apache Hadoop

**POPULAR NEWS REGARDING APACHE:**

The Apache Software Foundation [issued a notice](https://issues.apache.org/jira/browse/LEGAL-303?focusedCommentId=16088663&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-16088663) over the weekend, indicating that it has added [Facebook’s BSD+Patents license](https://code.facebook.com/pages/850928938376556) to its [Category X list of disallowed licenses](https://www.apache.org/legal/resolved#category-x) for [Apache PMC members](https://www.apache.org/dev/pmc.html). This is the license that Facebook uses for most of its open source projects.

**POPULARITY:**

In October 2012, 8,708 projects located at [SourceForge.net](https://en.wikipedia.org/wiki/SourceForge.net) were available under the terms of the Apache License. In a blog post from May 2008, [Google](https://en.wikipedia.org/wiki/Google) mentioned that over 25% of the nearly 100,000 projects then hosted on [Google Code](https://en.wikipedia.org/wiki/Google_Code) were using the Apache License,including the [Android operating system](https://en.wikipedia.org/wiki/Android_%28operating_system%29).

As of 2015, according to Black Duck Softwareand [GitHub](https://en.wikipedia.org/wiki/GitHub), the Apache license is the third most popular license in the [FOSS](https://en.wikipedia.org/wiki/FOSS) domain after [MIT license](https://en.wikipedia.org/wiki/MIT_license) and [GPLv2](https://en.wikipedia.org/wiki/GPLv2).

The [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD) project does not consider the Apache License 2.0 to be an acceptable license due to its patent provisions.