Open Source SW 이해와 실습

- Orientation -

지능로봇공학과 기석철

충북대학교 CHUNGBUK NATIONAL UNIVERSITY

E-campus / 강의자료
sckee@cbnu.ac.kr
E8-7, 341-1호

강좌 개요

- The course is a project course to **learn programming languages and conduct practice projects** over the semester. It deals with the **open-source tools** like GitHub and the tools are used in project management and troubleshooting.

1) Successfully explain the fundamental concept of open-source software
2) Practically use Linux system
3) Use various open-source software in software developments
4) Successfully identify the main differences in various open-source licenses

강좌 소개

강의

- 화요일 17:00 ~ 19:00 [E8-8, 305호]

수업 방식

- Open-Source Project
- Survey → Design → Implement → Present → Evaluate a project
- 3개 (A, B, C) 조별 수업
- 강의 관련 자료공유, 과제 제출 및 공지사항 등은 e-Campus 활용

평가

- 출석 : Pass / Fail
- Presentation Contents, Skill and Attitude : 20%
- Project Performance + Technical Report : 80%

강의 계획

- Source Code 와 보고서는 개인 Github 계정에 등록
- PPT 및 보고서는 e-Campus 과제물에도 제출

일자	학습 내용	발표 평가	제출물
09월 05일	Open Source 이해	-	-
09월 19/26일 10월 02일	프로젝트 계획	PPT	-
10월 17/24/31일	1차 중간 발표	PPT 동작 시연	-
11월 14/21/28일	2차 중간 발표	PPT 동작 시연	**중간 보고서**
12월 05/12일	최종 결과 발표	PPT 동작 시연	**결과 보고서**

What is Open-Source SW?

- Open-source software is computer software that has a source code available to the general public for use as is or with modifications. This software typically does not require a license fee.
- There are open-source software applications for a variety of different uses such as office automation, web design, content management, operating systems, and communications
- The key fact that makes open-source software (OSS) different from proprietary software is its license. As copyright material, software is almost always licensed. The license indicates how the software may be used. OSS is unique in that it is always released under a license that has been certified to meet the criteria of the Open-Source Definition. These criteria include the right to:
  - ① Redistribute the software without restriction;
  - ② Access the source code;
  - ③ Modify the source code; and
  - ④ Distribute the modified version of the software.

Is it free?

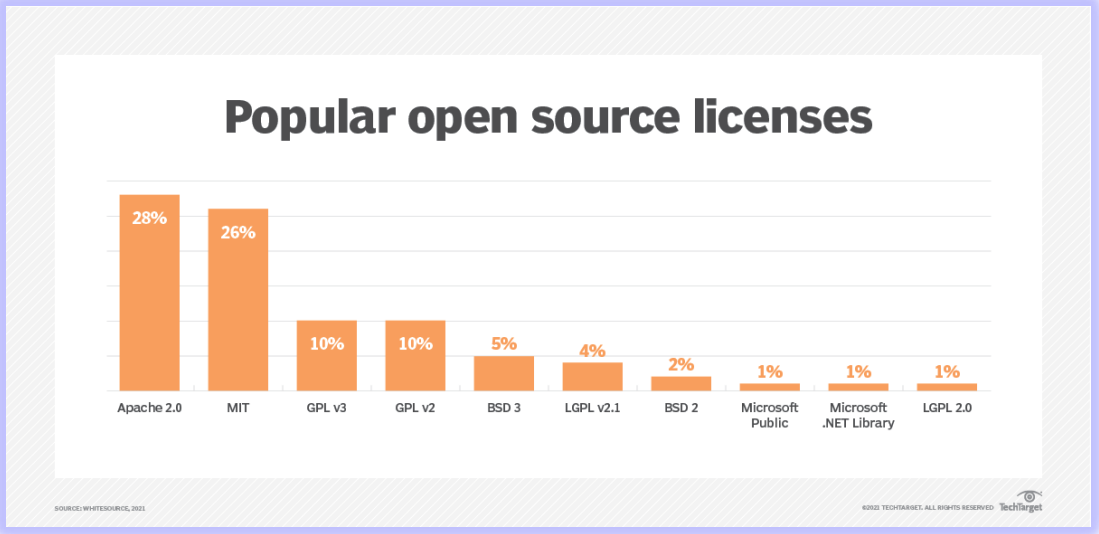
- Keep in mind that while OSS is usually free there are some exceptions. You will usually be able to determine what these exceptions are by considering the total cost of ownership (TCO) involved in adopting and managing open-source software. While the software itself may be free, make sure you consider the need for **additional services or products**, as these may have costs attached (e.g. access to software updates, support services).
- You also have to take into account possible **switching costs**. These costs would include moving data from an old system to new systems, training costs and costs involved when switching from one platform to another one (e.g. the costs of switching from Microsoft Windows to a Linux operating system). If your business does not have enough information technology expertise, you may have to outsource outside technical services to provide open-source support or to manage its implementation and delivery.

How did OSS get started?

- In the 1970s, proprietary software – i.e. software that did not allow users to redistribute it, modify it, or access its source code – became the norm.
- The development of open-source software was a reaction to the fact that changes or improvements could not be made to proprietary software by other developers or users. The open-source movement started with Richard Stallman's **general public license** model (in the 1980s), which holds that software should be freely modifiable, with the condition that if you make improvements to the software, you must put the improvements back in the open-source community. The rationale for the open-source movement is that a larger group of programmers not concerned with proprietary ownership will produce a better product.

GNU GPL

- The GNU General Public License, often shortened to GNU GPL (or simply GPL), lists terms and conditions for the copying, modification and redistribution of open-source software. The GPL was created by Richard Stallman to protect GNU software from being made proprietary.
- The GPL itself is under copyright notice by the Free Software Foundation (FSF), a tax-exempt charitable organization founded by Stallman's GNU project to generate funding for free software development.



Advantages of OSS

Lower Costs: Open source software usually does not require a licensing fee and its lower cost is generally one of the key reasons why small businesses choose to adopt this software. Make sure that you consider the total costs of ownership when considering open source software.

Flexibility: A programmer can take a standard software package and modify it to better suit business needs. You can usually hire a programmer to add a particular function to open source software.

Reliability and Quality: When looking at improved quality, you have to compare the products themselves. It is impossible to say that open source software is better than proprietary software in terms of reliability and quality – both have a range of products. However, mature open source software is generally viewed to be of good quality and reliability. If your business is not familiar with open source software, you may only want to review some of the more mature products (e.g. Linux, Apache and Sendmail).

Reduces "Vendor Lock-in": If you are using proprietary software you may be restricted to using certain vendors. Switching vendors in this case usually involves significant costs. Keep in mind though that choosing an OSS product may not make you totally independent of vendors. For some OSS products there may be a limited number of vendors that can provide you with services, upgrades or security patches.

Availability of External Support: External technical support is available for many of the open source software packages. Some vendors offer support contracts and there are service providers that install, configure and maintain an OSS system. Many open source products also have active online community support that may be able to answer your questions through online blogs.

Limitations of OSS

- **Lack of Personalized Support**. Unlike proprietary software, OSS packages do not come with phone support or personalized e-mail support. However, as mentioned there are commercial service providers who will provide support. If you need a lot of support, consider whether the overall costs of using an open-source product will be higher than that of a proprietary product.

- **Restricted Choice**. There are fewer choices available for open-source software.

- **Speed of Change**. Software is being modified on an ongoing basis in the open-source world, which can make it difficult to ensure that the software is compatible with other applications.

- **No warranty**. OSS does not come with a warranty, as there is no single company backing the product.

Examples of OSS

- You can find information about open-source products on the Internet by using a search engine and typing the keywords “open-source” followed by the “type of software application” that you are looking for. For example:
  ① Open-source web design.
  ② Open-source presentation software.
  ③ Open-source spreadsheets.
  ④ Open-source shopping carts.
  ⑤ Open-source communications (and so on).

- There are general directories that reference open-source applications such as www.wikipedia.com , www.opensourcecms.com , and www.cmsmatrix.org . These are good places to start your investigation. You may also want to look at various blogs and forums for product comparisons and reviews.

프로젝트 요구사항

- **팀 구성**
  - 1인 1조
  - 개인 단위로 평가함

- **주제 선정**

  자유 주제로 하되 다음 조건을 필히 준수하여야 함

  1) 프로그래밍 언어: Python (다양한 라이브러리 사용)
  2) 컴퓨터 운영체제: Ubuntu
  3) 개발 툴: Spyder, PyCharm, VS code 등

프로젝트 계획

1. 과제 제목 & 개요
    - 과제 내용을 명확하게 설명
2. 과제 필요성
    - 사회적, 기술적, 산업적 관점에서 필요성
3. 개발 목표
    - 프로그램 동작 시나리오
    - 정성적, 정량적 개발 목표
4. 프로젝트 설계
    - 개발 환경 구축
    - 화면 설계
    - Data, Function, Class 등 설계
5. 개발 일정

프로젝트 중간 발표

1. 과제 제목 & 개요
   - 과제 내용을 명확하게 설명
2. 과제 필요성
   - 사회적, 기술적, 산업적 관점에서 필요성
3. 개발 목표
   - 계획 대비 변경 항목
4. 프로젝트 설계
   - 계획 대비 변경 항목
5. 개발 일정 & 실적
6. 프로젝트 중간 결과
   - 진행 과정, 문제점, 향후 계획
   - 중간 결과 시연

프로젝트 결과 발표

1. 과제 제목 & 개요
    - 과제 내용을 명확하게 설명
2. 과제 필요성
    - 사회적, 기술적, 산업적 관점에서 필요성
3. 개발 목표
4. 프로젝트 설계
5. 개발 일정 & 실적
6. 프로젝트 결과
    - 진행 과정, 문제점 해결 방법
    - 결과 시연 및 성능 평가
7. 소감 및 후속 연구 계획

For this Class

GitHub

Ubuntu

ROS

OpenCV

OpenGL®

AutoWare

OPEN3D

apollo

CARLA

Open-source simulator for autonomous driving research.

Thank You !

충북대학교
CHUNGBUK NATIONAL UNIVERSITY