1. 오픈소스 소프트웨어

자유롭게 사용,복제,배포,수정 가능

소스코드가 공개되어 있는 소프트웨어

-Linux 커널

토발즈에 의해 최초로 개발

오늘날 가장 인기있는 운영체제 중 하나

현재까지도 개발중

-Apache

현재 가장 많이 이용되는 웹서버

-Bind

가장 많이 이용되는 DNS(Domain Name System)

호스트명을 IP 주소로 변환 시켜주는 프로그램으로 인터넷의 인프라를 구성하는 중요한 요소

-Mozilla Firefox

오픈소스 브라우저 프로젝트

1. 소프트웨어의 지적재산권과 라이센스

* 지적재산권법

-저작권

소프트웨어 개발 —> 컴퓨터프로그램저작권이 자동 발생 —> 프로그래머 or 회사에 부여

-특허권

하드웨어에 구현되거나 소프트웨어에 의해 동작이 구현되는 발명을 보호

자동 부여x,심사를 통해 부여되는 권리

특허권자의 명시적인 허락을 받아야함(오픈소스 소프트웨어,독점 소프트웨어)

-영업비밀

법에 의해 보호받고 있음

-상표

마케팅에 활용되는 이름 등을 보호

* 라이센스-권리자가 다른 사람에게 일정한 내용을 조건으로 하여 특정 행위를 할 수 있는 권한을 부여하는 권한

오픈소스 라이센스는 다음과 같이 사용자의 자유로운 사용,복제,배포,수정을 보장하고 있다.

-라이센시는 해당 오픈소스 소프트웨어를 자유롭게 사용할 수 있다.

-라이센시는 해당 오픈소스 소프트웨어를 자유롭게 복제할 수 있으며, 일정한 조건하에 재배포할 수 있다.

-라이센시는 해당 오픈소스 소프트웨어를 자유롭게 수정하여 사용할 수 있으며, 일정한 조건하에 수정된 내용을 재배포할 수 있다.

-라이센시는 해당 오픈소스 소프트웨어의 소스코드를 자유롭게 획득하고 접근할 수 있다.

오픈소스 라이센스에서 요구하고 있는 준수사항을 라이센시가 이행하지않으면 권리자로부터 소송 제기 가능

다운로드 시 라이센스의 요구사항 확인 필수

1. 라이센스의 구체적 내용

* 공통적 준수사항

-저작권 관련 문구 유지-소스코드를 임의로 수정하거나 삭제하여서는 안됨.

-제품명 중복 방지-오픈소스 소프트웨어와 동일한 이름을 제품명이나 서비스명으로 사용해서는 안됨.

-서로 다른 라이센스의 조합-오픈소스 소프트웨어에 다른 오픈소스 소프트웨어를 섞을 경우 반드시 두개의 라이센스가 서로 호환되는지를 확인

—— 여기서부터는 라이센스에 따라 다름 ———

-사용 여부 명시-소프트웨어를 사용할 때 해당 오픈소스 소프트웨어가 사용되었음을 명시적으로 표기

-소스코드 공개-수정하거나 추가한 부분이 있을 때 해당부분의 소스코드를 공개

-특허

* 라이센스별 준수사항

GPL 2.0

-현재 가장 많은 오픈소스 소프트웨어가 채택하고 있는 라이센스

-소프트웨어를 배포하는 경우 저작권 표시, 보증책임이 없다는 표시 및 GPL에 의해 배포된다는 사실 명시

-소프트웨어를 수정하거나 새로운 소프트웨어를 링크시키는 경우 GPL에 의해 소스 코드 제공해야 함.

-Object Code 또는 Executable Form으로 GPL 소프트웨어를 배포하는 경우, 소스 코드 그 자체를 함께 배포하거나 또는 소스코드를 제공받을 수 있는 방법에 대한 정보 함께 제공해야 함

-자신의 특허를 구현한 프로그램을 GPL로 배포할 때는 GPL 조건을 준수하는 이용자에게는 로열티를 받을 수 없으며, 제3자의 특허인 경우에도 특허권자가 Royalty-Free 형태의 라이센스를 제공해야만 해당 특허 기술을 구현한 프로그램을 GPL로 배포하는 것이 가능

LPGL(Lesser General Public License) 2.1

-소프트웨어를 배포하는 경우 저작권 표시, 보증책임이 없다는 표시 및 LGPL에 의해 배포된다는 사실 명시

-LGPL Library의 일부를 수정하는 경우 수정한 Library를 LGPL에 의해 소스 코드 공개

-LGPL Library에 응용프로그램을 링크시킬 경우 해당 응용프로그램의 소스를 공개할 필요 없음. 다만 사용자가 Library 수정 후 동일한 실행 파일을 생성할 수 있도록 Static Linking시에는 응용프로그램의 Object Code를 제공해야 함

-특허의 경우 GPL과 동일함

BSD(Berkeley Software Distribution) License

-GPL/LGPL보다 덜 제한적이기 때문에 허용범위가 넓음

-소프트웨어를 배포하는 경우 저작권 표시, 보증책임이 없다는 표시

-수정 프로그램에 대한 소스 코드의 공개를 요구하지 않기 때문에 상용 소프트웨어에 무제한 사용가능

Apache License

-BSD 라이센스와 비슷하여 소스코드 공개 등의 의무가 발생하지 않음

- "Apache"라는 이름에 대한 상표권을 침해하지 않아야 함

-특허권에 관한 내용이 포함되어 BSD 라이센스보다는 좀더 법적으로 완결된 내용을 담고 있음

MPL(Mozilla Public License)

-Netscape 브라우저의 소스코드를 공개하기 위해 개발된 라이센스

-소프트웨어를 배포하는 경우 저작권 표시, 보증책임이 없다는 표시 및 MPL에 의해 배포된다는 사실을 명시

-MPL 코드를 수정한 부분은 다시 MPL에 의해 배포

-MPL 코드와 다른 코드를 결합하여 프로그램을 만들 경우 MPL 코드를 제외한 결합 프로그램에 대한 소스코드는 공개할 필요가 없음

-소스코드를 적절한 형태로 제공하는 경우, 실행파일에 대한 라이센스는 MPL이 아닌 다른 것으로 선택가능

-특허기술이 구현된 프로그램의 경우 관련 사실을 ‘LEGAL’파일에 기록하여 배포

* 주요 쟁점

-소스코드 공개 여부,특허권,듀얼 라이센스

1. 기업에서의 오픈소스 라이센스 관리/활용 방안

오픈소스 소프트웨어 관련 정책을 수립할 필요성 증가

* 오픈소스의 장점

-Low Entry Cost:초기 개발을 시작하기 위한 비용이 적게 듬

-Fast,Flexible Development:독점 프로그램에 비해 기술 발전 속도가 빠름

-Open Format&Protocols:서로 다른 소프트웨어 간 상호 연동성이 보장됨

-Reliability & Stability:전세계에 있는 우수한 개발자들이 참여하기 때문에 독점 프로그램에 비해 비교적 안정적임

-Networking:다양하고 강력한 Networking 기능 지원

* 오픈소스의 단점

-애플리케이션의 부족:Linux에서 Windows에 버금가는 애플리케이션이 부족함 Linux-Windows 애플리케이션 호환x

-빈약한 문서:체계적인 문서를 갖지 못함

-불확실한 개발 로드맵:자발적 참여로 자유롭게 이루어지기 때문에 독점프로그램에서 볼 수 있는 로드맵을 기대하기 힘듬

-지적재산권-기업이 보유한 특허 및 소스코드를 포함시키려면 라이센스가 필요함

* 라이센스관리를 위한 프로세스/조직의 구축

개발이 끝난 이후에 오픈소스 라이센스관련 문제가 발견된다면 수정에 많은 시간과 비용이 소요됨--> 과제 계획 단계부터 오픈소스 라이센스문제를 고려할 필요가 있음

개발이 진행되면서도 단계별로 준수해야 할 사항들을 정의하고 반드시 체크해야만 함

'기획(SW Design)' -> '구현(Implementation)' -> '검증 (Verification)' -> '제품화(Production)'

기획

-Freshmeat.net, SourceForge, OSDir.com, BerliOS, Bioinformatics.org 등에서 해당 프로그램의 라이센스 확인

-소스코드 공개 여부 판단

구현

-소스 코드를 공개할 경우 회사보유의 지적재산권을 포함시키지 않도록 주의할 필요가 있음

-소스 코드를 공개를 원치 않을 시 다양한 문제가 생길 수 있으므로 라이센스 혹은 법률 전문가에게 의뢰하여 정확한 판단을 받아야 함

검증

-개발 계획서 그 자체로는 라이센스 이슈가 없었더라도 실제 구현 과정에서 개발자가 오픈소스 라이센스에 대한 검증없이 사용한 경우가 있을 수 있음

제품화

-사용된 오픈소스 소프트웨어들을 라이센스별로 분류

-각 라이센스에서 준수해야 할 사항들이 실제로 제품에 반영될 수 있도록 해야 함