**클라우드 #7주차 과제**

21660019 3A김세진

**1. metadata**

**2. tpm, tpmC**

**3. OSS**

**4. GPL, LGPL**

**5. IT compliance**

**6. BCP, DR**

**1. metadata**

1-(1) metadata의 개념

메타데이터(metadata)는 데이터에 대한 데이터이다. 이렇게 흔히들 간단히 정의하지만 엄격하게는 "어떤 목적을 가지고 만들어진 데이터 (Constructed data with a purpose)"라고도 정의한다.

1-(2) 내가 생각하는 metadata의 개념

메타데이터의 가장 쉬운 예시는 도서관의 라벨을 들 수 있는데, A~Z등의 앞글자에 따라서 인문, 기술, 사회, 예술 등의 분류가 되어 있는걸 봤을 것이다. 이렇듯 메타데이터라는 것은 어떤 정보를 분류, 파악하기 위한 데이터를 뜻한다. 1-(3) 메타데이터의 사례   
 실제로 내가 사용하고 있는 메타데이터는 여러가지가 있는데, 일단 자바프로그래밍을 하면서, //을 사용하여 컴파일에는 지장이 없는 주석을 이용할 때, 이 주석 또한 어떤 정보를 위한 코드(데이터)가 된다. 그리고 어노테이션 @~~ 의 경우에도 컴파일에는 지장이 없고 컴파일러에게 특정한 정보를 전달하는 기능을 한다.  
 자바를 벗어나서 html, xml과 같은 마크업 언어의 경우 <head></head>와 같은 블락을 형성하는데 이러한 표시 또한 이 부분이 어떤 기능을 하는지 알려주는 메타데이터이다.   
   
 현업에서 메타데이터는 거의 모든 곳에서 쓰일 것이다. 내가 작업한 모든 것을 다른 사람이 이해하기 어렵기도 할 것이기에 주석을 달아 놓는 경우가 많고, 프로그램 상에 오류를 체크하는 어노테이션이나 <>테그들도 정확성을 높여 주기 때문이다.

**2. tpm, tpmC**

2-(1) tpm, tpmC의 개념

tpm : (Transaction per minute)는 1분당 처리할 수 있는 트랜잭션의 처리를 나타내며, 이는 시스템의 성능을 나타내는 지표이다.

tpmC : TPC-C 벤치마크 시나리오에 대한 1분당 최대처리건수를 나타내는 수치로써,

DB기반 OLTP 미들웨어 시스템 서비스를 위한 H/W의 성능을 측정하는 대표적인 방법이다.  
  
2-(2) 내가 생각하는 tpm, tpmC의 개념

Tpm은 컴퓨터의 CPU가 작업을 처리하는 속도를 직접적으로 나타낸 지표이며, tpm이 높을수록, 계산이 빨라 트랜잭션의 처리가 많다는 것을 의미하는 것 같다. 하지만 실제로 하드웨어 제조사 마다 이런 Tpm의 성능을 측정하는 방식이 다를 수 있고, 실제로 사용할 때는 시스템마다의 오차로 다를 수 있기 때문에 TPC라는 기관에서 공식적으로 테스트 하여 온라인 상에서 실제 속도를 나타내는 지표는 tpmC라고 생각한다.

2-(3) BMT테스트

BTM(Benchmark Test)테스트는 특정 시나리오를 두고 어떤 하드웨어, 어떤 데이터베이스가 더 빠른지를 테스트하는 것이다.

실제로 하드웨어의 제조사는 해당하는 하드웨어가 몇 tpm의 속도를 낸다고 설명하지만, 실제로 시스템상에서 트랜잭션의 처리할 때는 어떤 상황인지에 따라서 속도가 다르기 때문에 직접적으로 알아내고 싶다면 각각의 하드웨어, 데이터베이스를 모두 시스템에 연결하여 알아야 하기에 이 테스트를 한다고 보면 될 것 같다.

**3. OSS**  
 3-(1) OSS의 개념

오픈소스 소프트웨어(OSS)는 소프트웨어의 설계도에 해당하는 소스코드를 인터넷 등을 통하여 무상으로 공개하여 누구나 그 소프트웨어를 개량하고, 이것을 재배포할 수 있도록 하는 것 또는 그런 소프트웨어를 말한다.  
  
 3-(2) 내가 생각하는 OSS의 개념  
 간단히 이해하면 이미 짜여졌던 프로그램 코드를 다른 사람들에게 무료로 드러낸 것을 오픈소스라고 하는 것 같다. 컴퓨터를 공부하면서 예전부터 생각한 것이 있다. 컴퓨터의 발전이나, 프로그램등의 발전이 우리 같은 학생의 정보 학습력보다 빠르게 발전하는 사회에서, 과연 우리가 배운다고 해서 현업에 그 발전력을 따라 갈 수 있을까라는 생각이었다. 우리가 아무리 배워도 기술은 더 발전하기에 과연 나아가는 발전에 나도 발을 들일 수 있을까 싶었다.   
 이런 상황의 사회에서 좋은 방법이 바로 오픈소스 인 것 같다. 발전된 기술이 아무도 모르게 꼭꼭 숨어 있다면, 우리 같이 뒤늦게 따라가는 사람은 따라가기에만 벅찰테지만, 열려진 기술이라면 그것을 따라서 쉽게 배울 수 있고, 활용할 수 있기 때문이다.   
 따라서 내가 생각하는 오픈소스는 초보자에게 빠른 가르침을 줄 수 있고, 개발자에게 더 좋은 방향으로의 발전을 제시할 수 있으며, 우리의 발전을 위해서 아주 좋은 방식의 운동인 것 같다.  
  
 3-(3) OSS의 발전사

1980년대 : 하드웨어 중심 체제에서 소프트웨어의 비중이 높아지면서 소프트웨어 기업의 소프트웨어에 대한 이용·배포·복제·수정 등에 일정한 제한을 가하려고 하는 추세가 있자, 이러한 ‘독점’ 체제에 반발해 ‘공유’를 주장하는 운동이 리차드 스톨만에 의해 시작

1990년대 : 1990년대부터는, 프로그래밍 소스 코드(source code)를 모두에게 개방한다고 해서 "오픈 소스(open source)"라는 용어가 생겨남.

2000년대 : Git허브가 생겨나면서 340만명의 사용자와 300만개의 코드 저장소가 이용됨. 가장인기있는 DVCS호스팅 사이트의 등장.

오픈소스는 한번 만들어진 소스를 다른 사용자가 보안을 더 해감으로써 버전이 업데이트 될 수 도있는데, 이러한 경우에 v0.0.1 -> v0.0.2이런 식으로 계속 발전해나가는 경향이 있다.

쉽게 말하면 요즘 나는 얼굴인식 도어락 프로잭트에 얼굴인식을 하는 face\_recognition이라는 파이썬 openCV의 오픈소스를 사용해야 하는데, 다른 사람의 이전 예시에서의 openCV의 버전은 2.4였고, 내가 가지고 있는 라즈베리와의 연동을 위해서는 3.4의 버전을 써야한다는 차이가 있어서 힘들었는데, 이처럼 오픈소스는 점차 기능이 추가되는 방향으로 발전함으로써 각 버전을 잘 업데이트 해야한다.

3-(4) 활용 방법, 사례

사례로는 iOS SDK를 들 수 있는데, 이 것은 애플의 앱 개발 프로그램이다. 핸드폰의 앱스토어에

가면, 개인이 만든 어플이 등록 되어 있는 것을 볼 수 있는데, 이것이 가능하게 한 것은 애플의

iOS SDK를 이용하여 iOS에서 구동되는 앱을 개발할 수 있게 되었기 때문이다.

이와 같이 기업이 자신의 소스를 대중에게 공개하여, 자신들이 생각하지 못한 어플들을 모든 사람들이 쓸 수 있게 공개하는 것이 오픈소스의 활용 방법이다.   
  
또 다른 방식의 사용 사례를 들자면, 이번에 나는 캡스톤 디자인이라는 과목을 듣고 있고, 거기서 라즈베리파이를 이용한 스마트 도어락을 만들려고 한다. 우리는 처음에 이것을 어떻게 만들까? 아직 어떻게 하는 지도 모르는데 얼굴인식은 어떻게 구현할까 걱정을 했다.

하지만 카메라만 있으면 파이썬 오픈소스로 이미 공개가 되있다는 것을 듣고, 다행이라는 것을 깨닳았다. 이처럼 우리 학생들의 공부에도 유용하게 사용 할 수 있다.

**4. GPL, LGPL**

4-(1) GPL, LGPL의 개념

GPL : GPL은 Free Software Foundation(FSF)에서 만든 Free 소프트웨어 라이선스이다.

소프트웨어의 실행, 연구, 공유, 수정의 자유를 최종 사용자에게 보장하는 라이선스이다.

LGPL : LGPL은 약소한(Less) GPL으로, 가장 큰 차이점은 LGPL 코드를 정적(static) 또는 동적(dynamic) 라이브러리로 사용한 프로그램을 개발하여 판매/배포할 경우에 프로그램의 소스코드를 공개하지 않아도 된다는 점이다. LGPL 코드를 사용했음을 명시만 하면 된다.

4-(2) 내가 생각하는 GPL, LGPL의 개념  
 1번의 오픈소스의 문제점을 해결하기 위해서 만들어진 라이선스라고 생각한다. 오픈소스의 문제점이라고 하며, 공짜라는 인식이 강하기에 자기 마음대로 개발하고, 그것을 판매함으로써 이득을 취하는 것이다. 따라서 이러한 행위에 대해서 어느정도 규제를 하는 것이 맞다고 생각하며, 이러한 규제를 하는 라이선스가 GPL, LGPL이라고 생각한다.  
   
  
4-(3) 현업에서 어떻게 사용할지

현업에서의 예시로는 한글과 컴퓨터를 예로 들 수 있는데, 한글과 컴퓨터측은 GPL이 적용된 Artifex사의 PDF 인터프리터 고스트스크립트를 적용했지만, 저작권자에게 비용을 제시하거나, 해상 모듈이 적용된 전체 프로그램의 소스 코드를 공개하지 않아서 법적으로 고소를 당했고, 해당 고스트 스크립트를 삭제하였다.

이렇듯 우리가 오픈소스를 사용함에 있어서 그저 학습용이나, 실습용의 상업적 목적이 아니라면, 아무렇게나 사용할 수 있지만, 상업용으로 사용하려면, GPL 라이센서의 규정을 지켜야 한다. 규정은 간단한데, GPL라이선스를 가진 코드를 사용했다면, 해당 프로그램의 소스를 오픈소스로 열어줘야 한다는 것이다.   
 이러한 방식은 더 여러 사람이 이 소스를 알게 해서 향후 발전에 기여하자는 식으로 많이 쓰이는 것 같다.

**5. IT compliance**

5-(1) IT compliance란?

Compliance란 준수, 따름이라는 의미인데, 이 앞에 IT가 붙었다.

대부분의 기업의 환경이 IT기반으로 바뀌어감에 따라 IT에 대한 의존도가 높아졌는데, 각종 침해, 해킹, 정보유출 등의 사고가 발생하기 시작했다. 따라서 이러한 위협으로부터 기업과 고객의 정보 자산을 보호하고, 기업 경영의 투명성을 높이기 위해서 IT관련 법, 제도등이 등장했고, 이러한 법과 제도를 준수하는 것이 IT 컴플라이언스라고 통칭한다.

5-(2) 내가 생각하는 IT compliance의 개념

예전에 조사한 ISMS, ISMS-P과 관련된 용어같다고 생각한다. ISMS와 ISMS-P는 사용자의 정보를 수집한 기업이 지켜야 할 정보통신망법, 개인정보보호법인데, 이는 기업이 정보를 유출할 시에 대한 법적 처벌을 다룬다. IP compliance란 이런 법을 지키는 방법, 혹은 지키는 것을 의미하는 용어 인 것 같다.

5-(3) 내가 클라우드 담당자라면 확인할 IT compliance.

조사하면서 여러 정보법을 알게 되었는데, 기존의 개인의 정보를 보호하는 것 이외에도, 회사의 부정부폐를 방지하기 위해 기업의 회계 원문을 보관해야하는 법이나, 인사, 노무, 안전 등에 관한 정보 보관, 관리 등의 법등 총 15,000여 가지의 법제가 존재한다고 한다.

내가 클라우드 담당자라고 했을 때, 이러한 모든 법제를 지킨다는 것은 불가능 할 것이기 때문에 가장 크게 강조되는 것과 필요한 것을 생각하여, 지키는 것을 생각할 것 같다.

따라서 주요하게 개인정보 보호, 회사의 정보 관리, 주요 정보 보안으로 생각이 들 것 같다. 가장 기본적으로 고객들의 정보를 확실히 보호하고, 정부에서 조사할 수 있게, 투명한 운영을 위한 회사 정보 관리, 마지막으로 회사에서 필수적으로 필요한 데이터를 보안유지 해놓는 방식으로 IT compliance를 준수 할 것 같다.

**6. BCP, DR**

6-(1) BCP, DR의 개념  
 BCP(business continuity planning) : 업무 연속 계획이라고 하며, 기업이 재해로 타격을 입은 뒤 업무 운명을 어떻게 복구 재개하는지에 대한 계획을 말한다. DR을 포함하는 상위 개념이다.  
 DR(Disaster Recovery) : 각종 재난 및 위험요소 들로 인한 정보시스템 장애 발생시 본래의 정상 기능으로 회복 시키는 모든 절차와 행위를 일컬어 재해 복구라고 한다

6-(2) 내가 생각하는 BCP, DR의 개념  
 DR이라는 것은 서버가 터지거나 어떤 불가피한 일로 인해 시스템이 정지되었을 때, 그것을 복구하는 것을 DR이라고 보면, BCP는 재해가 일어나기 전에 재해가 언제 발생할지 예측하고, 재해가 발생하면 어떻게 대처할지, 되돌리는 것은 어떻게 할지 등에 관한 모든 것을 계획해두는 것이라고 생각한다. 간략히 보면, BCP는 예비훈련을 한다고 생각하고, DR은 실제 상황이 터졌을 때, 향후 대책이라고 생각한다.   
  
 6-(3) 현업에서 어떻게 쓰일지

우리는 현재 COVID-19로 인해 여러 곤란한 상황을 겪었다. 업무를 못하는 회사도 있고, 집에서 업무를 하는 회사도 있고, 망해버리는 회사도 있다. 이러한 상황 속에도 우리는 경험이라는 것을 쌓는다. 다음부터 이런 상황에는 철저히 마스크를 쓰고, 방역을 하고, 사람간의 간격을 떨어트리고, 미리 화상회의 시스템을 마련하거나 기타 등등의 모습을 갖출 수 있다. 이렇듯 BCP는 우리가 겪은 것을 통해 다음 번에는 같은 상황 속에도 기업이 운영이 되게끔 그 시스템을 갖추는 방식으로 쓰일 것 같다.

6-(4) 모의 훈련 방법

1. 훈련의 필요성을 찾는다.

예) 최근 기업대상의 DDos 공격이 늘어났다. 최근 랜선웨어 감염 방지에 필요성이 대두됬다.

1. 훈련 시나리오를 짠다.

예) 랜선웨어가 A회사원의 컴퓨터에서 발생, 전사의 컴퓨터에 전염됬다.   
DDos공격으로 서버가 과부화되어 터지기 직전이다.

1. 훈련과정을 지정하고, 각 훈련과정마다의 결과를 평가한다.

예) 랜선웨어의 최초 발생지 탐색 (30분 , B등급)  
 전체 피해 상황 파악 (1시간, A-등급)

대체 사이트 운영 (30분, A등급)   
 복구 방법, 복구 (2시간, B등급)

총 소모 비용, 인원 확보 (30분, B등급)

재난 이후 관리

이런 식으로 각 기업마다 여러 상황을 생각하여, 시나리오를 짜고, 필요한 절차에 대한 시간, 과정을 기록하여 평가하는 방식으로 진행되며, 실제로 해당하는 일이 일어나면, 훈련으로 다져진 과정을 통하여 빠른 복구에 도움이 되고 손실이 적어 진다.

모의 훈련의 결과 평가지의 예시는 다음과 같다.

