교과목포트폴리오

20200820 김한수

알게 된 내용

- 아두이노
- 깃과 깃허브

아두이노

- 2005년 이탈리아에서 학생들을 대상으로 고안함.
- 다수의 스위치나 센서로부터 값을 받아들여 장치들을 통제.
- 쉽게 동작 가능.

팀 프로젝트를 통해 배운 아두이노

- 블루투스를 이용한 수중펌프 동작

```
/* motor */
if(blue == 0) {
  Serial.print("블루투스 켜기 값 : " + blue);
  if(sensorVal >= 650) {

    Motor(HIGH,255);//150의 출력으로 정방향 회전
    delay(2000);    //3000ms 즉 3초간 대기

    Motor(HIGH,0);//정지
    delay(1000);
  }
} else if (blue == 1) {

    Serial.print("블루투스 끄기 값: " + blue);
    Motor(LOW,0);//정지
    //delay(1000);
    Serial.println(blue);
}
Serial.println();
```

```
void Motor (boolean DIR, byte Motorspeed) { //펌프의 모터를 제어를 위해 정의한 함수, Motor(HIGH 또는
  analogWrite(PWMpin, (DIR) ? (255 - Motorspeed) : Motorspeed);
  digitalWrite(DIRpin, DIR);
}
```

팀 프로젝트를 통해 배운 아두이노

- OLED로 원하는 이미지 출력.

```
if(sensorVal >= 650) {
  display.setTextSize(3);
  display.setTextColor(WHITE);
  display.setCursor(17,5);
  display.println("T ^ T"); //display.println("T ^ T")
  display.display();
  delay(500);
  display.clearDisplay();
} else {
  display.setTextSize(3);
  display.setTextColor(WHITE);
  display.setCursor(17,5);
  display.println("*'u`*"); //display.println("T ^ T")
  display.display();
  delay(500);
  display.clearDisplay();
}
```

팀 프로젝트를 통해 배운 아두이노

- 블루투스로 연결된 기기에 센서값 전송.

```
// 블루투스 값
Serial.println(btSerial.read()); // 켜는게 48 끄는게 49
delay(1000);

  /* to android through bt */
  btSerial.print((String)jodo);
  btSerial.print(",");
  btSerial.print((String)sensorVal);
  btSerial.print(",");
  int temp = (int)CalcTemp(_rawObject);
  btSerial.print((String)temp);
  btSerial.print(",");
  delay(1);
```

팀 프로젝트를 통해 배운 아두이노

블루투스 정보를 읽는 기본 속도가 1초라는 것을 알게 되었다.

처음에는 모르고 코딩을 했다가 블루투스가 정보값을 읽기 전에 센서값이 계속 바뀌어서 오류가 발생했다.

블루투스 정보를 읽는 속도를 0.05초로 수정해서 오류 해결.

```
Serial.setTimeout(50);
//블루투스 통신 속도
```

팀 프로젝트를 통해 배운 아두이노

팀 프로젝트를 진행하며 여러 오류들을 경험하게 되었고 관련 정보를 찾아보며 아두이노 라이브러리, OLED 초기 변수 설정, 블루투스 통신값 변수 처리 등 여러 정보들을 배울 수 있었다.

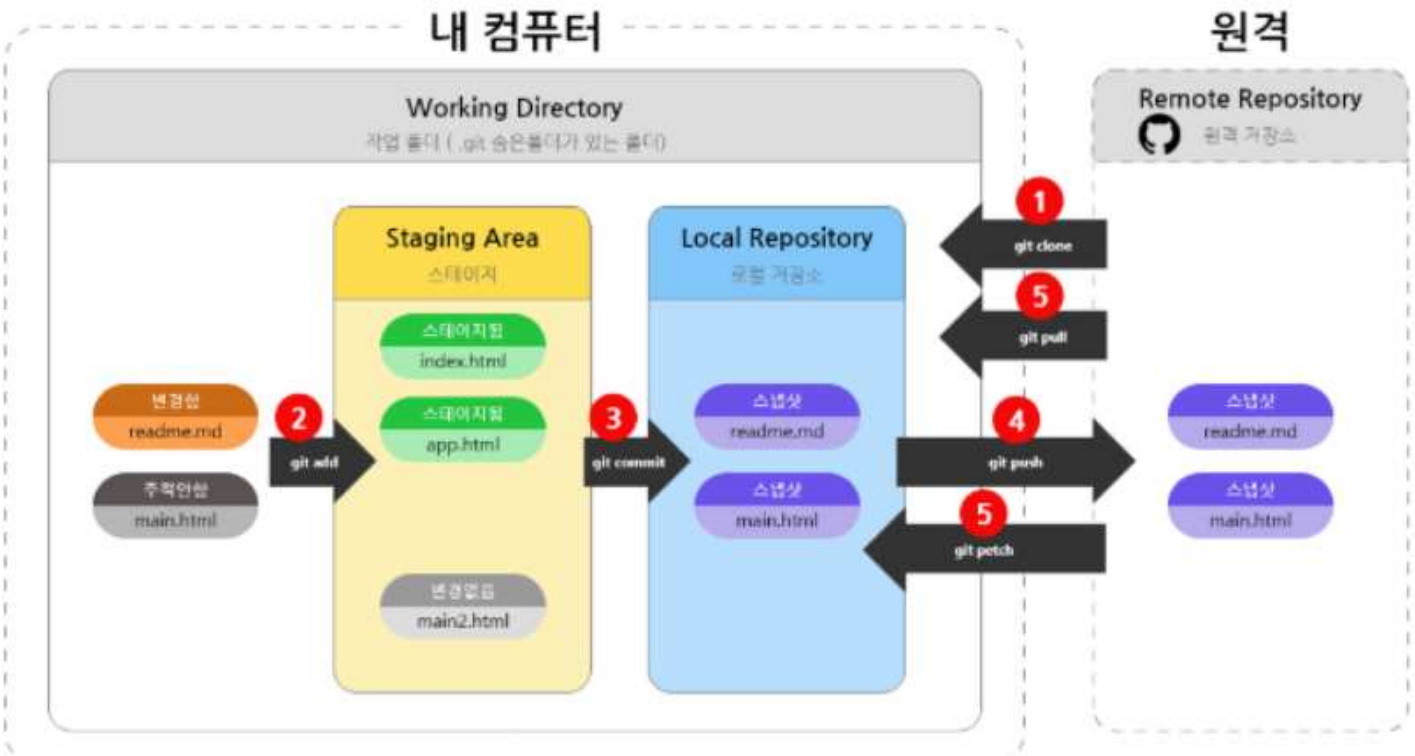
깃과 깃허브

깃허브
깃 (Git)
- 분산 버전 관리 시스템
- 협업 도구
- 소스의 버전관리 tool
- 무료 소수 관리 시스템

깃과 깃허브

깃허브 (Github)
- 깃의 저장소 및 관리 서비스
- 세계 최대 규모의 Git 저장소

깃의 구조


내 컴퓨터
원격
Working Directory
Remote Repository
Staging Area
Local Repository
index.html
app.html
main2.html
readme.md
main.html
git clone
git pull
git add
git commit
git push
git fetch
1
2
3
4
5
5
readme.md
main.html

버전관리

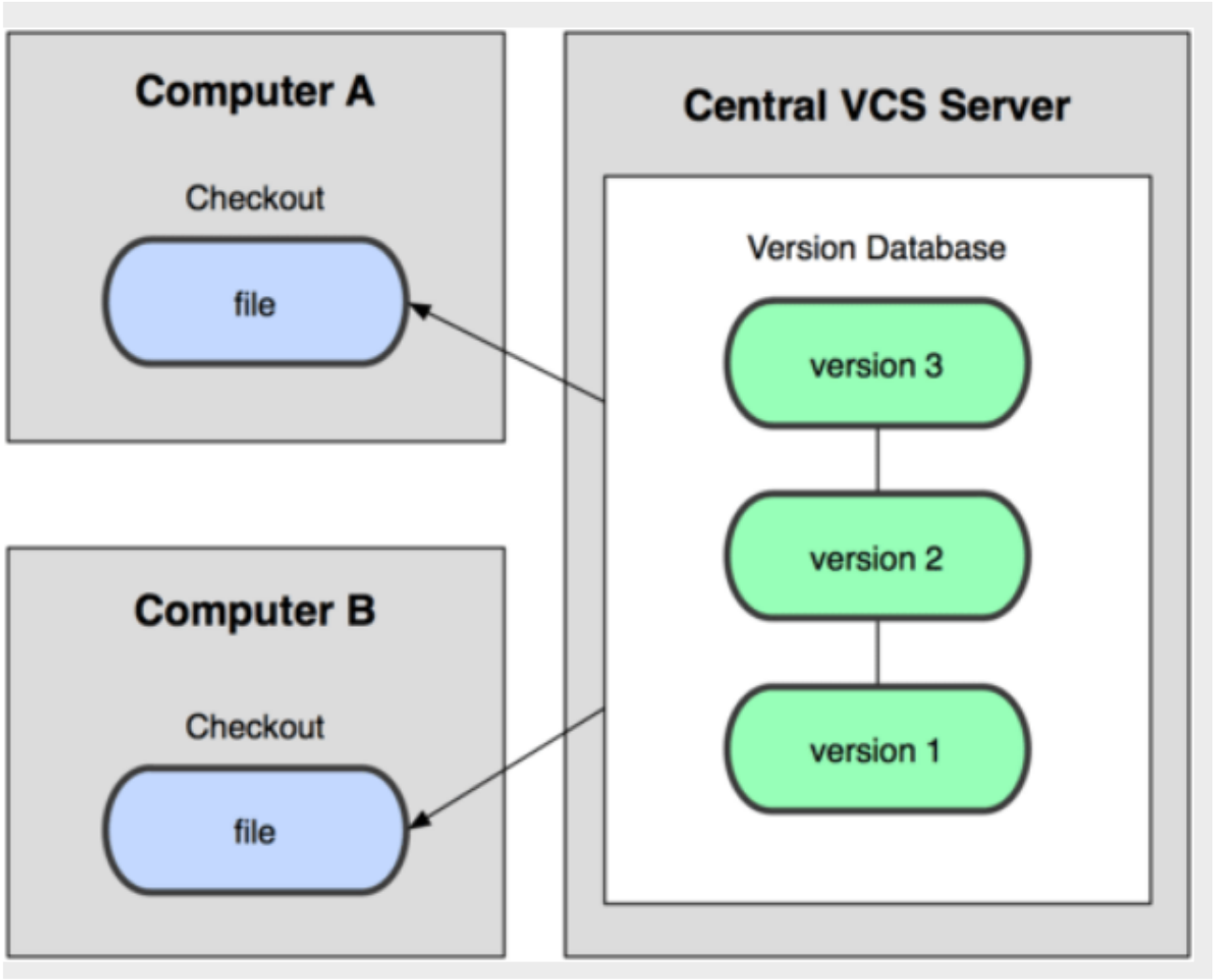
- 파일의 수정 이력 관리
- 여러 파일의 수정 묶음 관리

형상 관리

- 소프트웨어의 변경사항을 체계적으로 추척하고 통제하는 것
- 일반적인 단순 버전관리 기반의 소프트웨어 운용을 좀 더 포괄적인 학술 분야의 형태로 넓히는 근간

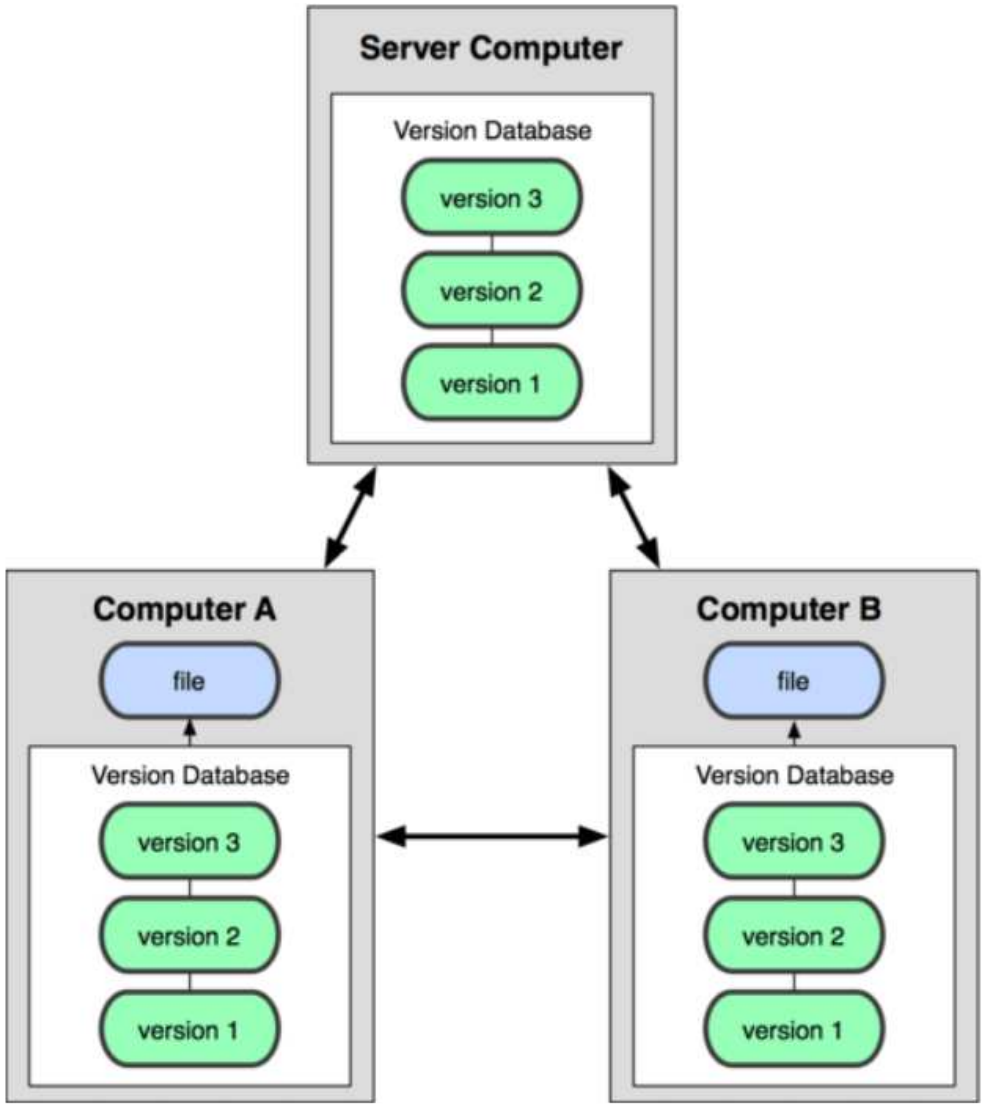
중앙집중식 버전 관리 시스템

- Subversion, CVS, Perforce 같은 CVCS 시스템은 많은 클라이언트가 중앙 서버로부터 파일을 받아서 사용
- 그런 이유로 중앙 서버의 하드디스크에 문제가 생겨서 자료가 날아가면 복잡해짐.


Computer A
Checkout
file
Computer B
Checkout
file
Central VCS Server
Version Database
version 3
version 2
version 1

분산형 버전 관리 시스템

- Git, Mecurial, Bazaar, Darcs 같은 DVCS 시스템은 클라이언트가 마지막 Snapshot을 받아오지 않고 그냥 저장소로 전부 복제
- 서버에 문제가 생기면 아무 클라이언트의 복제물로 서버를 복원 가능


Server Computer
Version Database
version 3
version 2
version 1
Computer A
file
Version Database
version 3
version 2
version 1
Computer B
file
Version Database
version 3
version 2
version 1

원격 저장소와 로컬 저장소

원격 저장소(Remote Repository)

- 파일이 원격 저장소 전용 서버에서 관리
- 여러 사람이 함께 공유하기 위한 저장소

로컬 저장소(Local Repository)

- 내 PC에 파일이 저장되는 개인 전용 저장소

CLI vs GUI

- GUI 프로그램의 대부분은 Git 기능 중 일부만 구현하기 때문에 비교적 단순
- CLI를 사용할 줄 알면 GUI도 사용할 수 있지만 반대는 성립하지 않음
- GUI를 사용하고 싶더라도 CLI가 기본으로 설치되는 도구

Git log 명령어

- commit history 확인

- Author 영역의 이름과 이메일 주소
- git config 명령을 통해 세팅했던 값 표기

Git add 명령어

- 작업 디렉토리 상의 변경 내용을 스테이징 영역에 추가하기 위해 사용하는 git 명령어

Git commit 명령어

- 명령어를 통해 명시적으로 기록을 남기기 전까지는 아무리 git add 명령어를 많이 실행해도 Git 저장소의 변경 이력에는 어떤 영향도 주지 않음

Git status 명령어

- Git add 명령어를 사용할 때, 항상 함께 사용되는 명령어
- 작업 디렉토리와 스테이징 영역의 상태를 확인하기 위해 사용

$ git init 명령어

- 새로운 저장소를 만들 때
- 다른 저장소의 URL을 이용해 저장소를 복사

$ git show [commit] 명령어

- 특정 커밋에 포함된 변경 사항과 메타데이터를 표시.

Discard Local change (로컬 변화 취소하기)

- 작업하고 있는 staging area나 working directory에서 작업하는 내용을 초기화하는 방법

$ git reset 명령어

- 커밋을 아예 취소
- Reset은 시계를 다시 맞추는 것
- 돌아가려는 커밋으로 리파지토리는 재설정 됨
- 해당 커밋 이후의 이력은 제거

$ git revert

- 현재 HEAD를 특정 시점의 commit 상태로 변경하는 commit을 추가로 수행
- 과거의 모든 커밋은 그대로 유지 관리되며
- 새로운 커밋을 추가해 측정 시점으로 이동