현장실습 주간보고서			
성 명	[redacted]	전공/학과	전기컴퓨터공학부
실습기간	20200706~20201224	학 번	[redacted]
실습기관명	[redacted]	실습기관 관리자	[redacted]

구분		실습사항(교육 및 실습내용)
1주	월 (2020.9.28)	vbs 네트워크 구조도를 도식화하는 작업을 진행하였다. 실제로 구조도를 그리니 잘못 알고있던 개념들 있었다. 예를들어 터미널이 참여한 네트워크를 구축할때 터미널에 따로 올더러 서버도 올려야하지만 이전 개념에서는 올더러를 생각하지않고 터미널서버만 추가하였다. 그리고 터미널 노드들의 볼륨 스토리지를 조절하였고 백업 서버를 4000기가 만들어 놨었는데 블록체인의 개념을 생각하니 백업이 필요하지 않을것같아 100기가만 남기고 다시 가격책정을 하였다.
	화 (2020.9.29)	클라우드 스펙과 가격에 따른 표를 제작하였다. case 1,2,3으로 나눈후 각각 터미널 모두참여, 터미널 5개참여, 터미널 미참여 시나리오를 생각하여 인스턴스를 분배하였다. 인스턴스 타입은 각 m2.c8m16,r2.c8m64, x1.c32m128을 사용하였고 볼륨 스토리지도 peer sever는 250GB만 할당하였다. 가격대는 각 사천칠백만원,사천만원,3900만원으로 나뉘었고 표 작성을 마무리하였다.
	수 (2020.9.30)	휴무일 ※ 실습일지는 1) 하루일정, 2) 업무내용, 3) 특이사항 등을 상세하게 기입할 것
	목 (2020.10.1)	휴무일
	금 (2020.10.2)	휴무일
2주	월 (2020.10.5)	배차서비스 관련하여 블록체인 네트워크를 구축하였었는데 이번에 그 네트워크에 이벤트리스너를 두기로 하였다. 채널로 연결되어있다면 한쪽 조직에서 데이터를 입력하거나 하면 다른 조직에 원장에도 그 데이터가 추가되는데 이때 추가는 되지만 api입장에서 언제 추가되는지 알수가없었다. 따라서 sdk를 이용하여 이벤트리스너를 만들어 원장이 바뀔때 api에게 알려주는 방식을 사용할것이다. 이를 위하여 document를 찾아보며 channelevnent listener를 만들기 위해 공부하였다.
	화 (2020.10.6)	event listener에 대해 공부하였다. 조직간에 생성된 채널을 통해 한조직이 업데이트를하면 다른조직도 자동으로 정보가 바뀌는데 이때 수동적으로 바뀐 조직은 언제 어떤 데이터가 바뀌었는지 손쉽게 알아보는 방법이 필요했다. 따라서 한 조직의 데이터가 바뀔때마다 event listener를 통하여 다른 조직에 알려주는 sdk를 이용하여 구현하였다. 그리고 회사서버에 만들었던 블록체인 네트워크를 옮기는 작업을 진행하였다.
	수 (2020.10.7)	회사 서버에 블록체인 네트워크를 옮겼다. remote-ssh라는 툴을 이용하여 서버에서 작업도 쉽게 vscode를 통해 연동하였다. 처음 서버에 옮긴후 api 컨테이너를 구축할때 npm error가 자꾸 떴는데 로그를 확인한 결과 network문제였다. dns서버를 찾는데 시간이 오래걸려 생기는 에러로 인해 이미지가 빌드되지 않았고 고치기위해 노력한 결과 docker build에 network host 옵션을 줌으로써 에러가 더이상 나지 않게되었다. 그리고 postman을 이용하여 서버에 쿼리한 결과 정상응답이 나오는것을 알수있었다.
	목 (2020.10.8)	세미나 준비한것을 발표하였다. 운영체제의 여러 기능중 가상화와 병행성에 대하여 자료조사를 하고 발표를 하엿는데 특히 cpu 가상화 부분에서는 스케쥴링을 중점적으로 발표하였다. 스케쥴링중 대부분의 운영체제에 현재 쓰이는 Multilevel Feedback Queue에 대해서 설명하였고 메모리가상화의 방법중 페이징과 세그먼트에 대해 발표하였다. 병행성 부분에서는 lock을 이용하여 임계영역을 지키는 것에 대해 발표하였다.
	금 (2020.10.9)	휴무일 ※ 주말에 근무한 경우, 대체한 날짜일지란에 [~일 대체근무]라고 표시하고 일지 작성(10.17일지 참고)
3주	월 (2020.10.12)	만든 블록체인 네트워크를 회사서버에 올리는 과정에서 새로운 오류가 발생하여 디버깅을 진행하였다. 도커 컨테이너는 생성되었지만 내부 채널이 생성되지 않아 체인코드설치를 실패하였는데 통신과 관련한 오류로 확인하였다. 처음 디버깅 할때는 서버의 권한때문에 파일이 생성을 되지만 읽지는 못하여 생기는 오류인줄 알았지만 로컬환경에서 똑같이 실행하니 정상 작동하는것을 알수있었다.
	화 (2020.10.13)	〔17일 토요일 대체근무〕 ping 명령어가 컨테이너에서 작동하지 않아 curl 명령어를 이용하여 컨테이너간의 통신을 진행하였다. 우선 로컬에서 curl 명령어를 이용하여 컨테이너에 요청을 보내니 empty resonse가 응답으로 돌아왔다. 하지만 서버에서 curl 명령어를 이용하여 컨테이

		너에 요청을 보내니 connection refuse가 나왔다. 컨테이너 간의 통신이 안되고 이때문에 카우치와 orderer간의 통신도 안된다는것을 알수있었다.
	수 (2020.10.14)	어떤 통신 오류가 블록체인 네트워크에서 발생하는지 찾는데 시간을 할애하였다. 로그를 분석한 결과 orderer사이의 통신이 제대로 이루어지지 않아 raft 합의가 이루어지지 않는 것을 확인하였다. orderer가 제기능을 하지않아 채널을 생성할때 블록을 이어 붙일수없어 네트워크가 제대로 구동되지 않았다. 저번 네트워크를 옮길때는 api를 설치할때 도커 이미지 빌드 과정에서 생기는 문제였는데 이번에는 그 이전 도커컨테이너간의 통신이 문제인것을 확인하였지만 고치지는 못하였다.
	목 (2020.10.15)	orderer와 peer간의 연결과 peer와 couchDB간의 연결이 안되는 것을 확인하였다. no route to host라는 로그를 계속 보게 되었는데 인터넷에 찾아보니 방화벽때문이라는 의견이 많았다. 하지만 서버의 방화벽을 함부로 건드릴수는 없어 더 진행하진 않았다. 그리고 ping 명령어로 직접 컨테이너의 응답을 받는 것은 가능하였는데 이를 보아 방화벽때문인것은 아닌것같다.
	금 (2020.10.16)	curl -IL옵션을 주어서 통신을 계속 진행하였다. 구글을 찾아보며 여러 질문글들을 보며 같은 에러가 있는지 확인하였고 그중 방화벽관련한 이야기가 가장 많은 것을 확인하였다. 하지만 컨테이너간의 통신인데 방화벽이 작동한다는것이 믿음직스럽지 않은 정보였고 회사 서버의 방화벽을 마음대로 조작하면 안될것 같아 방화벽쪽은 손대지 않았다. 로컬의 네트워크를 다시 서버로 옮겨보았지만 여전히 똑같이 실패하였다.
4주	월 (2020.10.19)	서버에 네트워크를 이전시키는것 외에 새로운 잡을 받았다. 저번에 만들었던 블록리스너를 이용하여 실제로 각 조직에 트랜잭션을 전달하는 함수를 만드는 것이다. 블록리스너에서 받아온 블록 json에서 키값과 밸류값을 파싱하여 빼내는작업을 하였고 이를 각 조직에 잘 전달되는지 확인하였다. 하지만 알수없는 이유로 각 조직에 전달이 잘될때가 있고 잘안될때가 있어 이를 고쳐야했다.
	화 (2020.10.20)	블록이벤트 리스너에서 블록을 생성할때마다 저번블록과 이번에 생성된 블록 두개가 로그로 찍히는 것을 분석하였다. 실제 orderer의 로그를 보면 입력할때마다 블록을 하나 생성하지만 블록이벤트 리스너에는 두개씩 로그가 나왔다. 그리고 채널로 연결된 각 조직에서 데이터를 입력할때 상대조직의 리스너에는 뜨지않지만 상대조직에서 블럭을 쌓을떄 같이 보이는 것을 확인하였는데 이때 어떤 로직으로 이런 현상이 나오는지 좀더 분석해야한다.
	수 (2020.10.21)	블록이 생성될때 어떨때는 한 조직에서 두개 어떨때는 한개라는것을 알게되었다. orderer는 블록을 하나 만들지만 이전데이터에 대해서 블록이벤트리스너가 계속 두개씩 가져오는 현상에 대해서 공부하였고 채널이벤트리스너나 트랜잭션이벤트 리스너에 대해서도 구글에서 관련 자료들을 찾아보았다. 그리고 이전 ITT에서 쓰던 코드를 받아 해당 코드를 분석하고 그 코드에 이벤트리스너를 구현하는 잡을 받았다.
	목 (2020.10.22)	그동안 invoke내부에 eventlistener를 구현하였었는데 게이트웨이를 끊는과정에서 이벤트리스너도 같이 꺼지는 것을 확인하고 이벤트리스너를 밖으로 빼내게 되었다. 이후 이벤트리스너가 제대로 동작 하기위해 route나 함수를 염두에 두며 구현하였고 이벤트리스너 기능이 제대로 되는것을 확인할수있었다. 현재는 console.log로 이벤트리스너를 띄우지만 다음은 해당 데이터를 받으면 api에서 또다른 기능이 실행될수있는 방향으로 할 예정이다.
	금 (2020.10.23)	블록이벤트 리스너에 대해서 검사를 받고 ITT사업에서 새로 쓰일 체인코드를 작성하였다. 기존의 체인코드를 참조하여 진행하였는데 nill pointer exception이 자꾸 발생하였다. 원인을 알수없어 우선 에러가 나는 지점을 찾아냈고 해당 함수에 들어가기전에 nill pointer를 string으로 바꾸어 실행하였다. 우선 postman으로 정상동작하는 것을 확인하였고 이제 nill pointer의 원인을 찾아 디버깅할 예정이다.
5주	월 (2020.10.26)	블록체인 기본개념에 대해 다시 공부하였다. 블록체인의 특징과 합의 그리고 기존시스템에 블록체인을 적용하였을때의 trade-off에 중점적으로 생각하였다. 기존의 퍼블릭 블록체인과 enterprise 블록체인인 하이퍼레저 패브릭을 비교하였고 각자의 장단점을 확인하였다. 그리고 하이퍼레저 패브릭을 사용했을때의 장점과 단점에 대해서 보고 어떤 서비스에 하이퍼레저 패브릭을 사용할수있는지 생각하였다.
	화 (2020.10.27)	카우치디비의 데이터를 제이슨으로 바꾼후 엑셀로 변환하였다. ittc 채널의 카우치디비에 쌓인 정보들을 제이슨 형식으로 바꾼후 엑셀형식으로 바꿨어야했는데 처음에는 변환사이트를 사용하였다. 하지만 데이터량이 많은 제이슨 파일에 대해서는 변환사이트에서 작동하지 않아 직접 코드를 짜서 csv파일로 만들었다. 이때 중첩 제이슨의 경우 기존의 변환코드로는 되지 않았기에 normalize를 사용하여 csv형식으로 바꾸었다. 이때 vscode에서 툴로 이 기능을 지원한다는 것도 알게되었다.
	수 (2020.10.28)	(31일 토요일 대체근무)요즘은 크롤링을 할때 beautifulsoup가 아닌 셀레니움 같은 도구를 사용한다는 것을 알게되었다. 크롤링을 할때 직접 파이썬으로 beautifulsoup라이브러리를 이용하였지만 속도가 느렸고 다른 개선점이 있는 것을 확인하였다. beautifulsoup로 진행할때는 이미지를 크롤링하는 속도가 초당 2개정도였다면 셀레니움을 이용하면 굉장히 빠른속도로 편하게 크롤링을 진행할수 있었다.
	목 (2020.10.29)	웹 크롤링을 진행하였다. 다른 연구원분의 요청으로 이미지와 동영상을 주제에 상관없이 1000개씩 크롤링 했다. 크롤링 진행 방식은 파이썬을 이용하였고 100000img.net에서 크롤링 하였다. 이미지는 바로 크롤링을 진행하였고 동영상의 경우 짧은 길이의 동영상이 필요하여 동영상사이트에서 크롤링하는것이 아닌 gif형식의 파일을 1000개 모은후 mp4파일로 변환하는 과정을 거쳐 크롤링하였다.
	금 (2020.10.30)	방화벽에 대해서 공부하였다. 특히 리눅스의 firewalld에 대해서 명령어와 기능 방식에 대하여

		찾아봤다. firewalld는 허용된 ip주소를 제외하고 차단하고 있었는데 이때 내 도커컨테이너의 ip를 화이트리스트에 추가하여 방화벽문제를 해결하고자 하였다. 두가지 해결방법이 제시되어있었는데 첫번째에는 nftables를 iptables로 바꾸는 것이었고 두번째는 어떤 트러스트 존을 이용하는것이서 관련 글들을 찾아보았다.