스트리밍 기술을 활용한 Darwin 스트리밍 서버 기반의 VOD 서비스인 렉플릭스 발표를 시작하겠습니다.

**개요**

KODI와 Hello tv의 VOD 서비스를 벤치마킹하였습니다.

KODI는 다양한 운영체제와 플랫폼을 지원하는 장점이 있지만 카테고리별로 분류가 제대로 되어있지 않아 원하는 영상을 찾기 불편합니다.

Hello tv는 TV용으로 리모컨으로 조작해야하기 때문에 편리한 인터페이스를 가지고 있습니다.

벤치마킹결과 윈도우와 안드로이드에서 프로그램을 지원하고 영상을 카테고리별로 분류하여 사용자 관점에서 조작이 편한 프로그램을 만들기로 했습니다.

다시보기 또는 영화를 집에서 편하게 보고 싶어 하는이를 타겟으로 스트리밍 서버를 구축하여 용량이 큰 비디오 파일을 다운로드없이 실시간으로 시청가능한 프로그램을 제작하였습니다.

서버에는 영화정보 서버와 스트리밍 서버가 있고 클라이언트에는 웹, 관리자, 안드로이드와 PC가 있습니다.

먼저 관리자가 동영상 파일을 직접 스트리밍서버에 올려둡니다.

그리고 영상 정보를 업로드하면 스트리밍 서버의 루트 폴더에 있는 동영상파일이 힌팅 됩니다.

여기서 힌팅이란 스트리밍 정보를 동영상 파일에 포함시키는 것을 말합니다.

힌팅이 완료되면 동영상을 스트리밍할 준비가 다 됩니다.

안드로이드나 PC에서 http를 통해 영상 정보를 서버에게 요청합니다. 그러면 서버는 xml형식으로 영상 정보를 주고, 파싱을 통해 원하는 정보를 뽑아내어 사용합니다.

안드로이드나 PC에서 RTSP를 통해 영상을 요구하면 서버는 RTSP로 사용자에게 영상 스트림을 보내 줍니다.

**개발환경**

총 4주동안 프로젝트를 진행했는데 일주일은 설계, 나머지는 코딩을 하고 23일은 디버깅 24일에는 PT를 준비하고 25일 발표를 했습니다.

김민준 팀원은 안드로이드 앱을 맡았고, 조영수 팀원은 서버구축과, 관리자 웹, 동영상 플레이어 전체 화면을 맡았습니다. 그리고 저는 PC 응용 프로그램과 동영상 플레이어에서 자막과 볼륨조절을 맡았습니다.

동영상 서버로는 다윈 스트리밍 서버를 사용했고, 관리자 웹에서는 파이참 4.5, SQLite3, 장고 1.8을, 안드로이드는 안드로이드 스튜디오 1.2, PC응용프로그램에서는 비주얼스튜디오2013MFC를 사용했습니다.

테스트환경으로 크롬과 인터넷익스플로러 모질라 파이어폭스, 키켓4.4.2와 4.4.4로 API 19를 사용했고 윈도우 7과 8을 사용했습니다.

**핵심기술**

FFmpeg과 SDL오픈소스를 사용했습니다.

FFmpeg의 libavformat 라이브러리는 먹싱과 디먹싱을 하고, libavcodec은 인코딩과 디코딩을 하기위해 필요합니다.

디먹싱을 하여 파일로부터 비트스트림 데이터를 추출하고 적합한 코덱으로 디코딩을 하면 원본 데이터 영상을 얻을 수 있습니다.

그래픽 사운드 입력 디바이스에 대한 레이어 인터페이스를 제공하는 SDL으로 디코딩된 영상이나 음성 스트림을 직접 화면이나 스피커로 출력하는 역할을 합니다.

자막은 SMI파일을 사용합니다. Smi 파일에는 자막을 나타낼 시간과 자막 정보가 있습니다. &nbsp; 가 있을 때는 자막을 출력하지 않는 경우입니다. 자막을 꺼내어 문자를 영상으로 바꿔주는 랜더링 작업을 거친다음 자막영상을 스크린위에 덮어씌움으로써 자막을 출력합니다.

자막의 싱크를 맞추기 위해서 현재 자막이 나타나는 시간과 다음에 자막이 나타날 시간을 이용합니다.

영상의 시간이 두 자막 시간 범위 사이에 있으면 자막을 출력합니다.

자막을 가운데 정렬, 자막이 길면 자동 개행하도록 했습니다.

자막의 처음부터 너비를 더해갑니다. 너비를 더해서 영상의 너비보다 커지면 커지기 전까지의 문자열을 잘라내어 자막버퍼에 집어넣습니다.

잘라내고 남은 자막도 자막버퍼에 넣어줍니다.

자막버퍼의 마지막에 있는 자막부터 화면의 아래에서부터 출력해줍니다.

출력할때는 화면의 너비와 잘라낸 자막의 너비를 이용해서 가운데정렬을 해줍니다.

더블 버퍼링을 적용하지 않으면 하나 그릴 때마다 화면이 깜빡입니다. 더블 버퍼링을 적용하면 가상화면에 모두 그리고 난 다음에 실제화면에 한번에 옮겨주어 깜빡이는 현상을 줄여줍니다.

영상의 포스터를 출력하기 위해서는 두가지 스레드를 사용하는데, 하나는 서버로부터 이미지 파일을 받는 스레드이고, 하나는 저장된 이미지파일을 출력하는 스레드 입니다.

변수를 두어 이미지 파일이 다운로드 되기 전에 출력을 되지 않도록 저장이 됐는지 확인하고 출력을 합니다.

MFC에서는 TinyXml 파서를, 안드로이드에서는 XPath 파서를 사용해서 Xml에 있는 데이터를 추출하여 얻습니다.

안드로이드에서 네비게이션 드로어를 사용하는데 평상시에는 화면이 감춰져 있다가 버튼을 클릭하면 화면을 나타내어 안드로이드의 화면이 작은 단점을 보완합니다.

**시연**