기말프로젝트 보고서  
경북대 LMS 강의자료 자동 다운로드

시스템프로그래밍(003)

2021113585 원인식

목차

- 1. 표지  
- 2. 목차  
- 3. 개요  
- 4. 프로젝트 구조  
- 5. 전체적인 흐름  
- 6. 참고자료

개요

시스템프로그래밍과 다른 과목을 수강하기 전 준비할 때 강의자료를 특정 폴더에 다운로드받고 그 자료를 가상머신에 옮겨야하는 등의 과정이 개인적으로 귀찮다고 느껴졌습니다. 이 과정을 자동화할 수 있다면 삶이 더 편리해질 수 있겠다는 생각이 들었습니다. 그래서 시스템 프로그래밍 기말 프로젝트의 주제로 경북대 LMS 강의자료 자동 다운로드를 선택하고 개발하게 되었습니다

이번 프로젝트를 개발하면서 설정한 목표는 “Linux C로 제대로 된 프로젝트를 만들어보자” 였습니다. C언어로 프로그래밍을 입문하고 관심이 많아서 C언어의 문법과 구조, 메모리 관리에 대해선 모르는 게 없었지만 Visual Studio로만 C언어를 접하고 개발 언어를 다른 언어로 바꾸면서 C언어를 기반으로 하는 프로젝트의 구성과 패키지 관리, 에러 처리, 외부 라이브러리 사용 경험이 전무했고 한편으로는 빈 디렉토리를 하나의 제대로 된 프로젝트를 만들어보면서 직접 손수 경험해보고 싶었기도 했습니다. 그래서 시간이 많이 걸리게 되었고 이러한 이유로 제출 기한을 지키지 못해 교수님께 죄송하다는 말씀을 드리고 싶습니다.

이 프로젝트는 HTTP 요청이 주된 기능이기 때문에 HTTP 요청을 날릴 수 있는 curl의 라이브러리 형태, libcurl을 사용하였고 결과로 날아오는 XML과 json 형식의 문서를 처리하기 위해 libxml2, json-c 라이브러리를 사용하였습니다. 또한 HTML 문서에서 결과를 빠르게 찾아낼 수 있는 css 선택자를 사용하기 위해 html-xml-utils 패키지의 hxnormalize와 hxselect 명령어를 사용하였습니다. 빌드 도구는 make와 cmake를 사용하였으며 단위 테스트 라이브러리는 cmocka를 사용하였습니다.

경북대 LMS 강의자료 자동 다운로드 프로젝트는 라이브러리가 크게 knuutils, knustring, knuapi, knulms, knudef로 나뉩니다. knudef는 knulms가 주로 쓸 StudentSession과 SubjectSession 구조체를 정의하였습니다. knustring은 HTTP 응답 결과를 처리하기 위해 C언어의 문자열 라이브러리가 제공하지 못하는 여러 유틸리티 함수와 그 함수의 주체가 될 KnuString 구조체를 정의하였습니다. knuapi는 libcurl을 직접 사용해 경북대 lms 서버에 로그인, 로그아웃, 강의자료 탐색 등의 요청을 날리고 응답을 파싱해 읽기 좋은 형태로 반환하는 저수준 함수를 모아놓았습니다. knulms 라이브러리는 knuapi를 활용해 라이브러리 사용자가 사용하기 쉽고 의도적으로 움직이게끔 설계한 고수준 함수를 모아놓았습니다.

마지막으로 해당 프로젝트는 Github을 통해 확인할 수 있습니다.  
- https://github.com/Insik-Won/knu-lms-auto-download

프로젝트 구조

전체적인 흐름

1. ./app/main.c의 main 함수를 통해 진입합니다.  
2. main.c의 process\_args()와 fill\_essential\_arguments()를 통해 옵션 및 인자를 main.c의 전역변수에 할당합니다.  
3. ./lib/knudef.c에 정의된 StudentSession과 SubjectSession 구조체, 그리고 ./lib/knulms.c에 정의된 knulms\_XXXXX류 함수를 사용해 lms에 접속하고 과목을 선택한 뒤 다운로드할 강의자료를 선택합니다.

3-1. knulms\_XXXXX류 함수는 ./lib/knuapi.c에 정의된 knuapi\_XXXXXX류 함수를 사용합니다 이 두가지 유형의 함수의 차이점은 knulms\_XXXXXX는 StudentSession과 SubjectSession을 사용한 높은 추상성을 지니고 메모리 관리, 소유권 관리도 라이브러리가 맡는 반면 knuapi\_XXXXXX는 libcurl, libxml2, json-c를 직접 호출하는 낮은 추상성에 메모리 관리, 소유권 관리를 사용자에게 맡깁니다.

3-2. 문자열 버퍼 관리 맟 처리는 주로 KnuString 구조체와 KnuString\_XXXXXX류 함수를 사용합니다. 또한 hxselect나 hxnormalize, unzip, gunzip 명령어 호출과 기타 유틸리티 함수는 ./lib/knuutils.c에 정의되어 있는 것을 씁니다.

4. 프로세스를 포크해 자식 프로세스는 선택한 강의자료를 다운로드하게 하고 부모 프로세스는 타이머를 설정하고 자식 프로세스가 끝날 때까지 다운로드 텍스트를 출력합니다. 이때 만약 인터럽트를 발생시킨다면 다운로드한 강의자료가 있는 디렉토리를 청소하고 프로그램을 끝냅니다.

5. lms 접속을 로그아웃하고 다운로드 완료 텍스트를 출력합니다. 메모리 관리는 라이브러리가 담당하기에 메모리 관리는 따로 하지 않습니다.

+. 테스트는 ./tests/all\_test.c에 있는 main에서 시작합니다 ./tests/test\_XXX에 있는 각 단위 테스트 함수를 끌어다가 테스트를 시작합니다. 단위 테스트에 쓸 유틸리티 함수는 test\_util.c에 정의되어 있습니다.

참고자료

- make 한국어 강의: <https://modoocode.com/311>  
- cmake 한국어 강의: <https://modoocode.com/332>  
- cmocka 공식 사이트: <https://cmocka.org/>  
- libcurl 공식 사이트: <https://curl.se/libcurl/c/>  
- libxml2 공식 사이트: <http://xmlsoft.org/>  
- json-c 공식 문서: <https://json-c.github.io/json-c/json-c-0.15/doc/html/index.html>  
- html-xml-utils 데비앙 문서: <https://manpages.debian.org/stretch/html-xml-utils/index.html>  
- gzip man 문서: <https://linux.die.net/man/1/gzip>  
- unzip man 문서: <https://linux.die.net/man/1/unzip>