과제02. 인코딩과 글꼴

- 실습 과정 보고서를 Markdown으로 작성하고 PDF로 변환하여 hw02.pdf 파일을 제출하세요.

실습1. 인코딩

다양한 형식의 텍스트 파일

다음과 같은 이름으로 유니코드 인코딩, BOM 포함 여부, 줄바꿈 문자의 유형이 다른 텍스트 파일을 만드세요. 각 파일에는 동일하게 "한글"이라는 두 글자와 엔터 키를 눌러 한 번 줄바꿈을 한 내용을 작성하세요. 명령행에서 `file` 명령을 이용하면 각 파일의 유형을 확인할 수 있습니다.

```
$ file hangul-utf8-crlf.txt
hangul-utf8-crlf.txt: Unicode text, UTF-8 text, with CRLF line terminators
```

파일명	인코딩	BOM	줄바꿈
hangul-utf8-lf.txt	UTF-8	없음	LF
hangul-utf8-crlf.txt	UTF-8	없음	CR LF
hangul-utf8-bom-lf.txt	UTF-8	있음	LF
hangul-utf8-bom-crlf.txt	UTF-8	있음	CR LF
hangul-utf16le-lf.txt	UTF-16LE	있음	LF
hangul-utf16le-crlf.txt	UTF-16LE	있음	CR LF
hangul-utf16be-lf.txt	UTF-16BE	있음	LF
hangul-utf816be-crlf.txt	UTF-16BE	있음	CR LF

- `hw02` 라는 디렉토리를 만들고 위 파일들을 모두 넣으세요.
- 터미널 프로그램을 실행하고 Bash 셸에서 해당 디렉토리로 이동하세요.
- 이 디렉토리에서 `file *` 명령을 내리면 모든 파일에 대해 `file` 명령을 적용한 결과가 나타납니다. 이것을 보여주는 스크린샷을 보고서에 포함하세요.

HEX 코드

이 파일들을 텍스트 에디터에서 열어보았을 때 보이는 모습은 구별되지 않습니다. 하지만 실제로 저장된 바이트들은 다릅니다. HEX 코드로 실제로 저장된 형태를 확인할 수 있습니다. HEX 덤프는

Visual Studio Code에 확장 기능을 설치하여 확인할 수도 있지만 명령행에서 `od` 명령을 이용하여 확인해 봅시다. 보고서에서 각 파일의 HEX 코드는 스크린샷을 이용하지 말고 Markdown의 코드 블럭을 이용하여 직접 입력하여 작성해 주세요. 예를 들면 다음과 같습니다. "한"이라는 글자는 `ed 95 9c`, "글"이라는 글자는 `ea b8 80`으로 각각 3바이트로 인코딩되어 있습니다. 줄바꿈 문자는 LF (`0a`)입니다.

```
ed 95 9c ea b8 80 0a
```

- 앞에서 작성한 각 파일의 HEX 덤프를 제시하세요.
- 두 글자 "한글"의 HEX 코드를 확인하세요.
- HEX 덤프에서 어느 부분이 BOM인지 확인하세요.
- HEX 덤프에서 어느 부분이 줄바꿈 문자인지 확인하세요.
- `od` 명령에 조사해 보고 실습을 위해 어떻게 사용했는지 명시하세요.

명령행 도구들

다음 명령들의 사용법을 조사하고 실제로 사용한 예시를 포함하여 보고서를 작성하세요.

- `file` : 파일 유형 정보 확인
- `iconv` : 인코딩 변환
- `dos2unix`, `unix2dos` : 줄바꿈 문자 형식 변환
- `bomstrip` (Linux): BOM 제거. UTF-8에서는 BOM이 필요없으며 BOM이 텍스트 처리에서 종종 불편을 야기하기 때문에 제거할 필요가 있는 경우가 있습니다.

각 명령의 사용법은 인터넷에서 검색해 보는 것도 좋습니다만 도움말을 꼭 확인해 보세요. 도움말은 `man` 명령 또는 `-h`, `--help` 옵션을 이용하여 볼 수 있습니다. 예를 들면 다음과 같습니다.

```
$ man file
$ file -h
$ file --help
```

실습2. 글꼴

텍스트 에디터에서 옛한글이 제대로 보이도록 글꼴을 설정해 봅시다. 이 실습은 옛한글 자체가 목적은 아닙니다. 중국어나 일본어의 한자를 다루거나 라틴 계열 문자를 사용하지 않은 여러 언어의 데이터를 다룰 때에는 글꼴도 문제가 되곤 합니다. 유니코드에 아무리 해당 문자가 정의되어 있다고 하더라도 글꼴이 해당 문자를 지원하지 않으면 컴퓨터 화면이나 프린터 출력에서 해당 문자가 시각적으로 나타나지 않기 때문입니다.

우선 글꼴을 설치해야 합니다. 옛한글을 포함하여 유니코드 문자들을 잘 지원하는 글꼴로는 한글과컴퓨터의 함초롬 글꼴에 기반해서 만들어진 글꼴, Adobe와 Google이 개발한 Noto 시리즈 글꼴 중 산돌커뮤니케이션에서 개발한 한글이 포함된 본고딕, 본명조가 있습니다.

- 한컴 다운로드: https://www.hancom.com/cs_center/csDownload.do
- 한글텍학회 함초롬LVT: http://ftp.ktug.org/KTUG/hcr-lvt/
- Noto fonts: https://fonts.google.com/noto/fonts
- 나눔옛한글: https://hangeul.naver.com/2014/archaicword

유니코드에서 한글을 표현하는 방법을 크게 두가지로 나누어 생각할 수 있습니다. 하나는 "한", "글"과 같이 모아쓰기를 하여 만들어진 하나의 음절 문자를 이용하는 것입니다. 다른 하나는 "ㅎ", "ㅏ", "ㄴ" 각각을 하나의 문자로 입력하고 화면에는 하나의 글자처럼 모아서 보이게 하는 것입니다. 전자의 방법은 모두 11172자의 글자를 이용할 수 있습니다. 후자의 방법은 160만이 넘는 조합이 가능합니다. 유니코드의 한글음절과 한글자모 영역을 확인하고 문자들을 사용하여 보세요. 운영체제들이 제공하는 문자표 프로그램을 이용하거나 graphemica.com 등의 사이트에서 검색하여 사용할 수 있습니다.

- Hangul Syllable
- Hangul Jamo

한글음절로 입력한 경우에는 한글을 지원하는 글꼴이라면 문제없이 잘 표시되지만 한글자모로 입력한 경우에는 제대로 표시해 주는 글꼴이 많지 않습니다. 위에서 설치한 글꼴을 이용해야 합니다.

- 옛한글이 포함된 적절한 텍스트 이용하세요. 예를 들어, 훈민정음 텍스트가 있습니다.
- 훈민정음언해 - 위키문헌, 우리 모두의 도서관
- Visual Studio Code에 옛한글이 포함된 텍스트를 붙여넣으세요.
- Visual Studio 설정에서 글꼴을 지정하여 옛한글이 제대로 표시되게 하십시오.
- 성공한 결과의 스크린샷을 보고서에 제시하세요.