

### Programming Assignment #3:

Implementation of the X-match compression algorithm in IA-32 assembly language

Due: November 28, 11:59PM. (~~November 21, 11:59PM.~~)

## 1. Introduction

이 과제는 파일 압축 함수를 IA-32 어셈블리 언어를 이용하여 구현하는 것이다. 이 과제의 목적은 무손실 압축기법의 일종인 X-Match 알고리즘 구현을 통하여 파일 압축방법에 대해 이해하고, IA-32 어셈블리 언어를 학습하는데 있다.

## 2. Problem Specification

### 2.1. Overview

다음과 같은 원형을 갖는 `xmatch2()` 함수를 IA-32 어셈블리 언어를 이용하여 구현한다. `xmatch2()` 함수는 첫 번째 과제로 구현했던 `xmatch()` 함수와 같이 2개의 `char` 포인터 (`input`, `output`)와 첫 번째 `char` 포인터가 가리키는 바이트열의 크기(`len`)를 인자로 받고, `xmatch2()` 함수 실행을 통해 얻은 압축된 파일의 크기를 반환한다.

```
int xmatch2 (char* input, int len, char* output);
```

`xmatch2()` 함수의 동작 방식은 첫 번째 과제에서 구현했던 `xmatch()` 함수와 같다.

첫 번째 인자인 `char* input`은 압축이 될 바이트 배열의 시작주소를 가리키며, 그 바이트열의 길이는 두 번째 인자인 `len`을 통해 입력된다. 이 때 입력되는 바이트의 크기는 4의 배수로 가정하고, 최대 크기는 65532를 넘지 않는 것으로 한다. (4의 배수가 아니거나,  $0 \leq len \leq 65532$  이 아닌 경우 음수를 리턴,  $len == 0$  인 경우에는 0을 리턴)

첫 번째 인자로 입력된 `input`을 X-Match 알고리즘을 통하여 압축하고 난 뒤, 세 번째 인자인 `output`이 가리키는 배열에 기록하고, 최종 압축 후의 바이트 수를 리턴 값으로 반환한다. 세 번째 `output` 인자가 가리키는 곳에는 `xmatch2()` 함수를 호출하기 전 최악의 경우 필요한 크기를 가정하여 메모리를 이미 할당해 놓았기 때문에 `xmatch2()` 함수 내에서 메모리 공간을 새롭게 할당할 필요는 없다.

`xmatch2()` 함수 내에서는 입력 받은 4바이트 \* N (N은 0 이상의 정수) 크기의 데이터를 4바이트 씩 잘라서 압축을 수행한다. (자세한 사항은 과제1 안내 문서의 2.2절을 참조)

함수의 실행 결과, 오류 없이 압축이 성공했을 경우 최종 압축 후의 바이트 수를, 오류가 발생했을 경우 음수를 반환한다.

본 과제 수행에 있어서 별도의 함수를 생성할 필요가 있는 경우, `xmatch2.s` 파일 내에 구현할 수 있다.

## 2.2. Restrictions

- `xmatch2()` 함수 내에서는 라이브러리 함수 사용이 불가능하다. 단, 필요에 따라 새로운 함수를 작성하여 사용하는 것은 무방하다.
- `xmatch2()` 함수의 `setup`, `finish` 코드를 제외한 메인 부분 코드 작성시, 사용 가능한 레지스터는 `%eax`, `%ecx`, `%edx` 총 3개로 제한한다.
- 과제는 리눅스 환경에서 수행하고, 과제 보고서에 리눅스 환경에서 컴파일하고, 실행한 화면을 캡처하여 추가한다.
- 정상적인 실행 결과를 리턴하는 과제 제출물 중 수행속도를 기준으로 상위 10%에 추가 점수를 부여한다.

## 3. Skeleton Codes

본 과제 수행을 위하여 아래와 같이 3개의 파일이 주어진다.

Makefile: GNU make utility를 위해 사용되는 파일.  
main.c: `xmatch2()` 함수를 호출하는 `main()` 함수  
xmatch2.s: `xmatch2()` 알고리즘을 구현해 넣을 파일

과제를 위해서는 `xmatch2.s` 파일만을 수정하면 된다. (`xmatch2.s` 외에 다른 파일을 추가하지는 말 것)

## 4. Hand in instructions

- 작성한 프로그램 코드 상단에 이름과 학번을 주석으로 넣는다.
- 과제 제출시 `xmatch2.s` 파일을 "학번.s" 파일로 이름을 변경하여 제출한다 (예: 2009310123.s)
- 프로그램의 설계, 구현에 관한 내용을 담은 보고서를 별도로 제출한다. 보고서는 워드, 한글 등의 형식도 상관 없지만 가능하면 PDF format으로 제출하고, 제출시 "학번.pdf" 형태로 파일을 저장하여 제출한다.
- 과제 제출은 프로그램 소스 코드인 "학번.c"와 보고서인 "학번.pdf"를 메일로 첨부하여 [sse2030@csl.skku.edu](mailto:sse2030@csl.skku.edu) 로 보낸다. 과제 제출시 메일의 제목은 아래와 같은 형식을 따른다.

[SSE2030] PA#3, 학번, 이름

## 5. Logistics

- 과제 제출 결과는 본 수업의 홈페이지인 <http://csl.skku.edu/SSE2030F10> 에서 확인할 수 있다.
- 과제 제출 시간은 메일 도착시간을 기준으로 하며, 기한을 넘겨 제출할 경우 하루에 25% 씩 감점한다.
- 과제에 대한 의논은 함께 할 수 있으나, 프로그램 소스코드 작성은 스스로 해야 한다.
- 다른 사람의 과제를 copy한 경우, 두 사람 모두 0점 처리한다. 인터넷 등에서 찾은 소스코드를 그대로 copy한 경우에도 0점 처리한다. 두 번 이상 이와 같은 이유로 0점 처리된 경우 F 학점을 받을 수 있다.

Good luck!

----

담당조교: 김형준, 031-299-4970, #400629 (hjkim at csl.skku.edu)