암호프리미티브구현

GMP의 자료형 및 기본 함수 실습

장남수

1. 데이터의 표현

데이터의 표현

- 데이터의 개념
  - 전송 받은 데이터
    - Assume that a=0Xaaaabbbb11112222.
- 텍스트 데이터
  - 문자열로 첫 번째 부터 순서대로 메모리에 저장
  - 대부분 8비트(octet,char) 단위를 기본으로 사용하여 표현

주소	0	1	2	3	4	5	6	7	8	9	10
데이터	A	s	s	u	m	e		t	h	a	t
Hex값	41	73	73	75	6d	65	20	74	68	61	74
주소	14	15	16	17	18	19	20	21	22	23	24
데이터		a	=	0	X	a	a	a	a	b	b
Hex값	20	61	3d	30	58	61	61	61	61	62	62
주소	28	29	30	31	32						
데이터	b	b	1	1	1	1	2	2	2	2	.
Hex값	62	62	31	31	31	31	32	32	32	32	2e

데이터의 표현

- 데이터의 개념
  - 전송 받은 데이터
    - Assume that a=0Xaaaabbbb11112222.
- 숫자 데이터
  - 기본 데이터 단위(8, 16, 32)로 나누어 하위부터 메모리에 저장
  - 대부분 32비트(unsigned long) 단위를 기본으로 사용하여 표현
    - 8비트인 경우

주소	7	6	5	4	3	2	1	0
Hex값	aa	aa	bb	bb	11	11	22	22

    - 16비트인 경우

주소	3	2	1	0
Hex값	aaaa	bbbb	1111	2222

    - 32비트인 경우

주소	1	0
Hex값	aaaabbbb	11112222

데이터의 표현

- ✓ 큰 정수의 표현
  - ✓ 기본 메모리(32비트)를 넘어가는 데이터를 표현해야 한다.
  - ✓ +, - 부호를 표현해야 한다.
  - ✓ 음수로 16진수 0x11111111222222223333333344444444을 표현하려면?
    - ✓ 32비트 메모리 4개와 부호를 표시하는 영역이 필요
- ✓ GMP의 큰 정수 표현을 위한 구조체 : mpz_t

gmp.h

```
typedef struct
{
        int _mp_alloc;          /* Number of *limbs* allocated and pointed
                                   to by the _mp_d field.  */
        int _mp_size;           /* abs(_mp_size) is the number of limbs the
                                   last field points to.  If _mp_size is
                                   negative this is a negative number.  */
        mp_limb_t *_mp_d;       /* Pointer to the limbs.  */
} __mpz_struct;
```

```
typedef unsigned long int       mp_limb_t;
```

데이터의 표현

- ✓ 큰 정수의 표현
  - ✓ 기본 메모리(32비트)를 넘어가는 데이터를 표현해야 한다.
  - ✓ +, - 부호를 표현해야 한다.
  - ✓ 음수로 16진수 0x11111111222222223333333344444444을 표현하려면?
    - ✓ 32비트 메모리 4개와 부호를 표시하는 영역이 필요
- ✓ GMP의 큰 정수 표현을 위한 구조체 : mpz_t

```
#include <stdlib.h>
#include <stdio.h>
#include "gmp.h"


void main()

{

    mpz_t a,b,c;

    a->
        _mp_alloc
        _mp_d
        _mp_size
}
```

데이터의 표현

- ✓ 큰 정수의 표현
  - ✓ 기본 메모리(32비트)를 넘어가는 데이터를 표현해야 한다.
  - ✓ +, - 부호를 표현해야 한다.
  - ✓ 음수로 16진수 0x11111111222222223333333344444444을 표현하려면?
    - ✓ 32비트 메모리 4개와 부호를 표시하는 영역이 필요
- ✓ GMP의 큰 정수 표현을 위한 구조체 : mpz_t
  - ✓ mpz_t 구조체로 선언된 a에 값을 넣으려면?

a->_mp_alloc : 할당한 배열의 개수
a->_mp_size : 실제 데이터가 들어있는 배열의 개수
a->_mp_d : 배열의 포인터

데이터의 표현

- ✓ 큰 정수의 표현
  - ✓ 기본 메모리(32비트)를 넘어가는 데이터를 표현해야 한다.
  - ✓ +, - 부호를 표현해야 한다.
  - ✓ 음수로 16진수 0x11111111222222223333333344444444을 표현하려면?
    - ✓ 32비트 메모리 4개와 부호를 표시하는 영역이 필요
- ✓ GMP의 큰 정수 표현을 위한 구조체 : mpz_t
  - ✓ mpz_t 구조체로 선언된 a에 값을 넣으려면?

```
a->_mp_d     = (unsigned long int *) malloc(20);  /*32비트 배열 5개 할당*/
a->_mp_alloc = 5;                                 /*할당된 배열개수 입력*/

a->_mp_d[0] = 0x44444444;                         /* 데이터 입력*/
a->_mp_d[1] = 0x33333333;
a->_mp_d[2] = 0x22222222;
a->_mp_d[3] = 0x11111111;

a_mp_size    = -4;                                /*(실제 데이터 개수 * 부호) 입력*/
```

데이터의 표현

- ✓ 큰 정수의 표현
  - ✓ 기본 메모리(32비트)를 넘어가는 데이터를 표현해야 한다.
  - ✓ +, - 부호를 표현해야 한다.
  - ✓ 음수로 16진수 0x1111111122222222333333334444444을 표현하려면?
    - ✓ 32비트 메모리 4개와 부호를 표시하는 영역이 필요
- ✓ GMP의 큰 정수 표현을 위한 구조체 : mpz_t
  - ✓ mpz_t 구조체로 선언된 a에 값을 넣으려면?

```
mpz_init(a);
...
mpz_set_str(a,"-1111111122222222333333334444444",16);
```

```
mpz_init_set_str(a,"-1111111122222222333333334444444",16);
```

2. GMP의 기본 함수

GMP의 기본 함수

- ✓ Initialization Functions
  - void mpz_init (mpz t integer)
    - 정수형 구조체를 초기화(배열 할당)하고 0을 입력
  - void mpz_init2 (mpz t integer, unsigned long n)
    - 정수형 구조체를 초기화(배열 할당)하고 0을 입력
    - 배열은 n비트의 데이터가 들어갈 수 있는 크기로 할당
  - void mpz_clear (mpz t integer)
    - 배열을 초기화(배열 해제)
  - void mpz_realloc2 (mpz t integer, unsigned long n)
    - 할당된 배열의 개수를 n비트의 데이터가 들어갈 수 있는 크기로 재할당
    - 기존에 데이터가 있는 경우 이를 유지

Gmp(지엠피)의 함수를 종류별로 몇가지 살펴보도록 하겠습니다. 가장 먼저 살펴볼 것은 정수 데이터를 입력하는 구조체인 mpz(엠피지)의 초기화 함수입니다. Mpz_init(엠피지 이니셜라이즈)함수는 구조체에 unsigned long int(언사인드 롱 인트)형의 배열을 1개 할당하고 값으로 0을 입력하는 함수로 가장 기본이 되는 함수입니다. Mpz_init2(엠피지 이니셜라이즈 투)함수는 구조체에 입력할 데이터의 길이를 아는 경우로 데이터를 입력할 때 메모리 재할당을 줄이기 위하여 데이터의 비트크기를 입력으로 받아 필요한 만큼의 배열을 할당합니다. 그리고 mpz_clear(엠피지 클리어) 함수는 값을 사용하고 나서 더 이상 필요 없는 경우 메모리 낭비를 막기 위하여 배열을 해제하는 함수입니다. 마지막으로 mpz_realloc2(엠피지 리얼록케이션 투) 함수는 할당된 배열의 개수를 n비트의 데이터가 들어갈 수 있는 크기로 재할당하는 함수로 기존에 데이터가 있는 경우 이를 유지합니다. 실제 gmp(지엠피)에서 제공하는 함수를 사용할 때 함수 내부에서 메모리가 부족하다고 판단되는 경우 realloc(리얼록케이션)을 하므로 사용하는 경우가 많이 발생하지는 않습니다.

GMP의 기본 함수

- ✓ Assignment Functions
  - void mpz_set (mpz t rop, mpz t op)
    - op를 rop에 복사
  - void mpz_set_ui (mpz t rop, unsigned long int op)
    - unsigned long int 형 데이터를 rop에 복사하고 size를 1로 할당 (양수로 인식)
  - void mpz_set_si (mpz t rop, signed long int op)
    - signed long int 형 데이터를 rop에 복사하고 부호에 따라 size를 1 또는 -1로 할당
  - int mpz_set_str (mpz t rop, char *str, int base)
    - 스트링 형태로 base 진법의 데이터를 입력받아 저장하는 함수
    - 예) 10진수 데이터 : mpz_set_str(a,"-123456789",10)
    - 예) 16진수 데이터 : mpz_set_str(a,"-abcd12345",16)
  - void mpz_swap (mpz t rop1, mpz t rop2)

다음으로는 assignment(어사인먼트) 함수에 대하여 살펴보겠습니다. 앞으로 설명할 함수에서 op(오피)는 오퍼랜드, rop(알오피)는 리턴드 오퍼랜드를 의미합니다. 따라서 다음 함수들에서 알 수 있듯이 gmp(지엠피)는 결과값 rop(알오피)가 항상 함수의 첫번째 변수로 정의합니다. Mpz_set(엠피지 셑) 함수는 op에 있는 데이터를 rop에 복사하는 함수입니다. 이때 rop에 할당된 메모리가 부족하면 자동적으로 realloc(리얼로케이션) 됩니다. mpz_set_ui (엠피지셑 유아이) 함수는 unsigned long int (언사인드 롱 인트)형 데이터를 rop에 복사하고 부호는 양수로 인식하여 size(사이즈)를 1로 할당합니다. mpz_set_si (엠피지 셑 유아이) 함수는 signed long int(사인드 롱 인트)형 데이터를 rop에 복사하고 부호에 따라 size를 1 또는 -1로 할당합니다. mpz_set_str (엠피지 셑 에스티알) 함수는 예제와 같이 스트링 형태로 base 진법의 데이터를 입력받아 rop에 저장하는 함수입니다. 다른 함수와 다르게 int(인트)으로 값을 반환하게 되어있으며 정상적인 스트링의 형태이면 0, 잘못된 형태이면 -1을 반환하게 되어있습니다. 마지막으로 mpz_swap (엠피지 스왑) 함수는 rop1(알오피원)과 rop2(알오피투) 를 바꾸어 내보내는 함수입니다.

GMP의 기본 함수

- ✓ Combined Initialization and Assignment Functions
  - void mpz_init_set (mpz t rop, mpz t op)
    - op의 데이터를 저장할 수 있는 만큼의 데이터를 rop에 할당하고 복사하는 함수
  - void mpz_init_set_ui (mpz t rop, unsigned long int op)
    - rop를 할당하고 op를 복사하는 함수
  - void mpz_init_set_si (mpz t rop, signed long int op)
    - rop를 할당하고 op를 복사하는 함수
  - int mpz_init_set_str (mpz t rop, char *str, int base)
    - 스트링 형태의 데이터를 저장할 수 있을 만큼의 배열을 할당하고 base 진법의 데이터를 입력받아 저장하는 함수

다음으로는 assignment(어사인먼트)와 초기와 함수의 기능을 동시에 하는 함수입니다. 각각의 rop가 초기화 되지 않은 상태에서 rop를 초기화하고 각각의 값들을 할당하는 역할을 합니다.

GMP의 기본 함수

✓ Conversion Functions

- unsigned long int mpz_get_ui (mpz t op)
    - 부호를 무시하고 op 0번째 주소 데이터를 반환하는 함수

- signed long int mpz_get_si (mpz t op)
    - 부호를 고려하여 op 0번째 주소 데이터를 반환하는 함수
    - 표현상에 문제를 일으킬 수 있음

- char * mpz_get_str (char *str, int base, mpz t op)
    - mpz(엠피지) 형태의 데이터를 스트링(문자열)으로 바꾸어주는 역할

컨버전 함수는 어사인먼트의 반대 기능을 하는 함수이며 이중에서 실제 가장 필요한 함수는 mpz_get_str(지엠피 겟 에티알) 함수로 mpz(엠피지) 형태의 데이터를 문자열로 바꾸어주는 역할을 합니다. 이때 base의 진법으로 값을 바꾸어 데이터를 반환하며 값은 숫자가 아닌 아스키형태로 변환됩니다.

GMP의 기본 함수

✓ Conversion Functions

- unsigned long int mpz_get_ui (mpz t op)
    - 부호를 무시하고 op 0번째 주소 데이터를 반환하는 함수

- signed long int mpz_get_si (mpz t op)
    - 부호를 고려하여 op 0번째 주소 데이터를 반환하는 함수
    - 표현상에 문제를 일으킬 수 있음

- char * mpz_get_str (char *str, int base, mpz t op)
    - mpz(엠피지) 형태의 데이터를 스트링(문자열)으로 바꾸어주는 역할

컨버전 함수는 어사인먼트의 반대 기능을 하는 함수이며 이중에서 실제 가장 필요한 함수는 mpz_get_str(지엠피 겟 에티알) 함수로 mpz(엠피지) 형태의 데이터를 문자열로 바꾸어주는 역할을 합니다. 이때 base의 진법으로 값을 바꾸어 데이터를 반환하며 값은 숫자가 아닌 아스키형태로 변환됩니다.

3. GMP의 기본 함수 사용 실습

GMP의 기본 함수 사용 실습

- ✓ 10진수 문자열 데이터 a, b를 입력받아 mpz로 변환 후 화면 출력
- ✓ a, b 데이터를 서로 바꾸고 16진수 스트링으로 출력
  - ✓ a : 12345678987654321
  - ✓ b: -98765432123456789