

# **báo cáo midterm project môn thực hành kiến trúc máy tính**

Họ và Tên: Trần Văn Tuấn  
MSSV: 20184223  
ID Nhóm: 19 – 1 thành viên  
Đề bài: 17

## 1. Phân tích bài toán

- Viết chương trình nhận vào 1 chuỗi bit và thực hiện chuyển sang ký tự ascii tương ứng.
- cứ mỗi 8 bit ứng với 1 ký tự
  - chiều dài của chuỗi bit phải là bội của 8
  - vì là chuỗi nhị phân nên chuỗi chỉ được xuất hiện ký tự '1' or '0'

## 2. Cách thực hiện

- ý tưởng ở đây là sẽ chuyển lần lượt 8 bit một sang ký tự ascii tương ứng và in ra màn hình cho tới khi kết thúc chuỗi (đương nhiên trước đó cần đọc được chuỗi bit và đảm bảo độ dài hợp lệ)
- quá trình chuyển như sau : 8 bit (nhị phân) → decimal (thập phân) → ký tự ascii
- để lấy được 8 bit lần lượt → cần cung cấp vị trí bắt đầu duyệt và tăng lên 8 đơn vị sau khi duyệt xong 8 bit đầu và cứ thế tới hết
- khi duyệt 8 bit thì sẽ check luôn xem ký tự có hợp lệ (tức bằng 1 or 0) hay không, nếu không thì dừng chương trình
- nếu ký tự = '0' thì bỏ qua còn = '1' thì cộng lũy thừa cơ số 2 với số mũ tương ứng để tìm decimal (cứ cộng dồn, hết 8 bit là ra decimal)
- nghĩ tới 1 hàm đệ qui (có thể dùng 2 for lồng nhau nhưng em dùng đệ qui)

- hàm tương ứng trong C:

```
void print_ascii(char *bitString, int viTriBatDau) {
    int decimal = 0;
    int j = 0;
    for (; j < 8; j++) {
        if (bitString[i+j] == '\0') return;
        if (bitString[i+j] == '0') continue;
        if (bitString[i+j] != '1') return; // != '0' and != '1' → error → return

        // xử lý trường hợp = '1'
        decimal = decimal + pow(7 - j); // chuyển sang hệ 10
        // hàm pow(n) trả về  $2^n$ 
    }
}
```

```

    // tới đây thì in ra màn hình ký tự tương ứng
    printf("%c", decimal);
    print_ascii(bitString, viTriBatDau + 8);    // tiếp tục gọi đệ qui
}

```

- mã triển khai trên mips thì mời Thầy xem trong file mã nguồn. Trong đó em có giải thích ý nghĩa của các thanh ghi cũng như các chương trình con

## Kết quả thực hiện

- input : 010010000110010101101100011011000110111100100001

- output : Hello!

- ảnh kết quả:

The screenshot displays the Mars MIPS simulator interface. The **Text Segment** window shows the following assembly code:

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24020008	addiu \$2,\$0,0x00000008	8: li \$v0, 8 #
	0x00400004	0x3c011001	lui \$1,0x00001001	9: la \$a0, bit_string # get bina...
	0x00400008	0x34240000	ori \$4,\$1,0x00000000	
	0x0040000c	0x24050100	addiu \$5,\$0,0x00000100	10: la \$a1, 256 # and stor...
	0x00400010	0x0000000c	syscall	11: syscall #
	0x00400014	0x0004a821	addu \$21,\$0,\$4	12: move \$s5, \$a0 #
	0x00400018	0x34010000	ori \$1,\$0,0x00000000	14: la \$a0, 0(\$s5)
	0x0040001c	0x02a12020	add \$4,\$21,\$1	
	0x00400020	0x0c100014	jal 0x00400050	15: jal count_char
	0x00400024	0x00028020	add \$16,\$0,\$2	16: add \$s0, \$0, \$v0 # \$s0 = le...

The **Data Segment** window shows memory addresses from 0x10010000 to 0x100100e0, all containing zero. The **Mars Messages** window shows the input string '010010000110010101101100011011000110111100100001', the output 'Hello!', and the message '-- program is finished running --'.