

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Kiến Trúc Máy Tính

Báo cáo Bài tập lớn 2

Đề tài 3:

Chuyển chữ hoa thành chữ thường và ngược lại, đối với các ký tự khác thì giữ nguyên. Chuỗi $A[N]$ là chuỗi nguồn, chuỗi $B[N]$ là chuỗi đích. $N = 1000$.

Giáo viên hướng dẫn: Nguyễn Xuân Minh

Danh sách các thành viên

STT	Họ và Tên	MSSV
1	Phạm Tấn Đại	1710929
2	Cao Thành Nhân	1710214
3	Lê Thị Thanh Thảo	1713177

Mục lục

I. Cơ sở lý thuyết	
1. Lý thuyết.	3
2. Flow chart.	4
3. Các bước giải thuật.	5
II. Hiện thực bằng hợp ngữ assembly MIPS5	
1. Giải thích code	6
2. Thời gian thực thi (Sau khi cải tiến)	12

I. Cơ sở lý thuyết

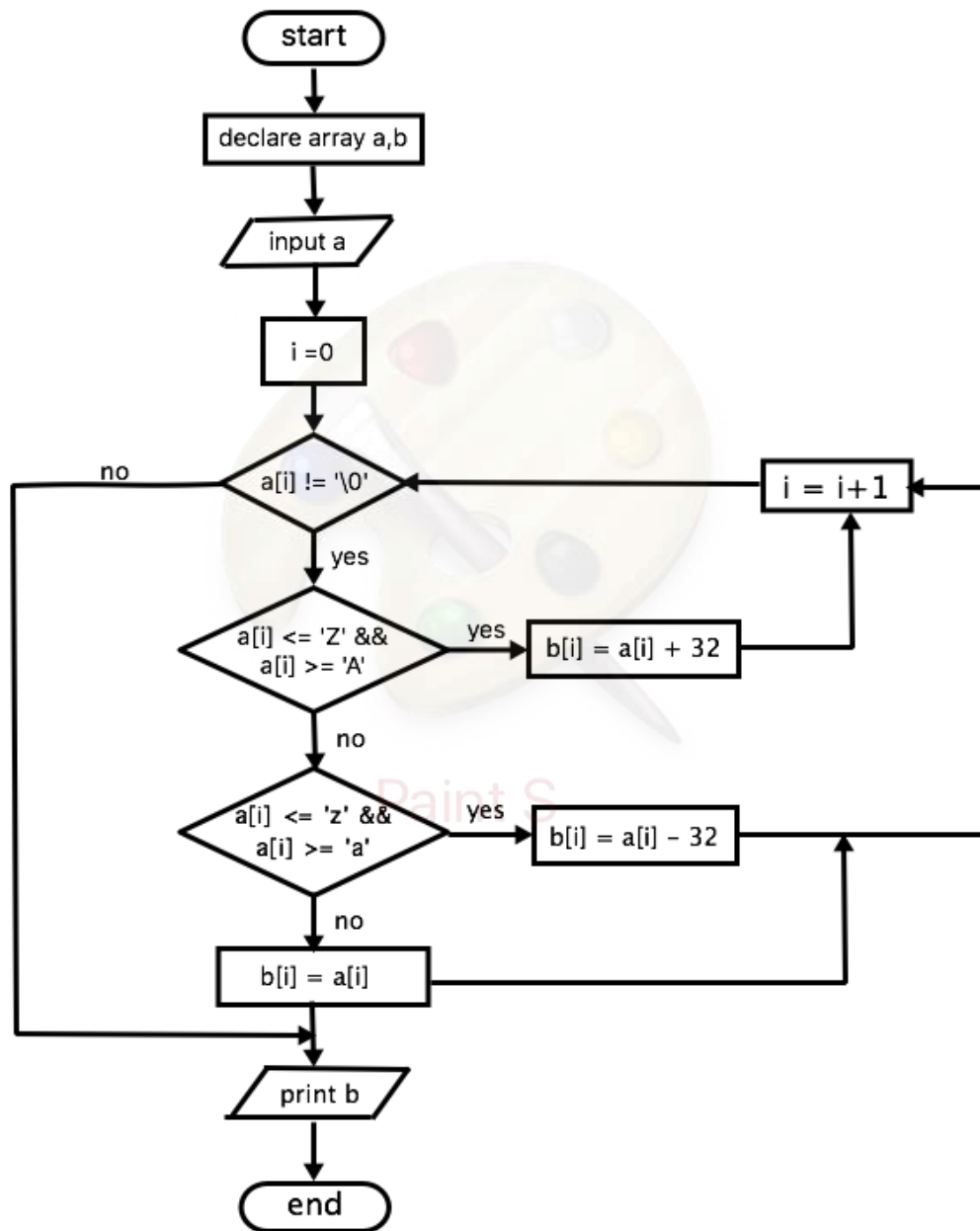
1. Lý thuyết

- Dựa vào bảng mã ascii để xem biểu diễn các kí tự ở dạng Decimal, từ đó việc chuyển đổi các kí tự hoa thành thường và ngược lại trở nên dễ dàng .

Decimal	Hex	Char	Decimal	Hex	Char
64	40	@	96	60	`
65	41	A	97	61	a
66	42	B	98	62	b
67	43	C	99	63	c
68	44	D	100	64	d
69	45	E	101	65	e
70	46	F	102	66	f
71	47	G	103	67	g
72	48	H	104	68	h
73	49	I	105	69	i
74	4A	J	106	6A	j
75	4B	K	107	6B	k
76	4C	L	108	6C	l
77	4D	M	109	6D	m
78	4E	N	110	6E	n
79	4F	O	111	6F	o
80	50	P	112	70	p
81	51	Q	113	71	q
82	52	R	114	72	r
83	53	S	115	73	s
84	54	T	116	74	t
85	55	U	117	75	u
86	56	V	118	76	v
87	57	W	119	77	w
88	58	X	120	78	x
89	59	Y	121	79	y
90	5A	Z	122	7A	z

- Để chuyển kí tự hoa thành thường , ta cộng thêm 32. Ngược lại, để chuyển kí tự thường thành hoa ta trừ cho 32.

2. Flow Chart



3. Các bước giải thuật

- Bước 1: **Load address** của chuỗi string vừa nhập vào trong **\$t0**
- Bước 2: Cho vào vòng lặp loop, dùng lệnh **lb** để lấy từng kí tự trong chuỗi string rồi lưu vào **\$t1** để xét , khi nào đến kí tự '\0' thì dừng vòng lặp (`exit_loop`)
- Dùng lệnh **blt** để so sánh **\$t1** với 'a' nếu nhỏ hơn thì nhảy đến case1. Dùng lệnh **bgt** để so sánh **\$t1** với 'z' nếu lớn hơn thì nhảy đến **case2** (tăng \$t0 lên 1).

Nếu \$t1 nằm giữa 'a' và 'z' thì thực hiện chuyển kí tự thường thành hoa bằng cách dùng lệnh **addiu**

```
addiu $t1,$t1,-32
```

- Nếu không phải kí tự thường thì nhảy đến **case1** để xét đến kí tự hoa. Dùng lệnh **blt** để so sánh **\$t1** với 'A', bgt để so sánh **\$t1** với 'Z' . Nếu **\$t1** nằm giữa 'A' và 'Z' thì thực hiện chuyển kí tự **hoa** thành **thường** bằng cách dùng lệnh **addiu**

```
addiu $t1,$t1,32
```

- ✓ Bước 3: Nhảy đến **case2** tăng \$t0 lên 1

```
addiu $t0,$t0,1
```

- ✓ Bước 4: Nhảy đến **loop** để thực hiện tiếp vòng lặp. Dùng lệnh **beqz** để so sánh **\$t1** với '\0', nếu bằng nhau thì dừng vòng lặp.

```
beqz $t1,exit_loop
```

- ✓ Bước 5: **Load address** của chuỗi string vừa chuyển vào trong **\$a0** và xuất kết quả ra console.

II. Giải thích code và thời gian thực thi:

1. Giải thích code:

***Chú thích:** Ý nghĩa màu sắc trong đoạn giải thích:

- text:** đoạn code tương ứng trong code C.
- text:** chức năng cơ bản của đoạn code trong cùng khung.
- #text:** giải thích từng câu lệnh

Code	Giải thích
<pre>\$LC0: .ascii "Nhap chuoi: \000" \$LC1: .ascii "Chuoi sau khi da chuyen la: %s\000"</pre>	Khai báo 2 chuỗi sẽ hiển thị khi nhập xuất dữ liệu.
<pre>main: addiu \$sp,\$sp,-2040 sw \$31,2036(\$sp) sw \$fp,2032(\$sp) move \$fp,\$sp</pre>	<pre>#Vào hàm main Dời stack point \$sp = sp - 2040, [\$sp + 2036] = \$ra #[\$sp + 2032] = \$fp #\$fp = \$sp</pre>
<pre>sw \$0,1028(\$fp) addiu \$2,\$fp,1032 li \$3,996 # 0x3e4 move \$6,\$3 move \$5,\$0 move \$4,\$2 jal memset nop</pre>	<pre>char a[1000], b[1000] = ""; Khởi tạo giá trị chuỗi. #\$0 = \$fp + 1028 #\$2 = \$fp + 1032 #\$3 = 996 #\$6 = \$3 #\$5 = \$0 #\$4 = \$2 #Thực hiện việc khởi o</pre>
<pre>lui \$2,%hi(\$LC0)</pre>	<pre>printf("Nhap chuoi: "); Hiển thị chuỗi yêu cầu nhập.</pre>

<pre>addiu \$4,\$2,%lo(\$LC0) jal printf nop</pre>	<pre>#\$4 = \$LC0 #jal đến hàm printf</pre>
<pre>lui \$2,%hi(stdin) lw \$3,%lo(stdin)(\$2) addiu \$2,\$fp,28 move \$6,\$3 li \$5,1000 # 0x3e8 move \$4,\$2 jal fgets nop</pre>	<pre>fgets(a, 1000, stdin);</pre> <p>Lấy kết quả nhập vào.</p> <pre>#\$3 = [stdin] (dữ liệu nhập vào) #\$2 = \$fp + 28 #\$6 = \$3 #\$5 = 1000 #\$4 = \$2 #jal đến hàm gets.</pre>
<pre>sw \$0,24(\$fp) b \$L2 nop</pre>	<pre>for (int i = 0; a[i] != '\0'; i++)</pre> <p>Vào vòng for.</p> <pre>#[\$fp + 24] = \$0 #Rẽ nhánh đến \$L2</pre>
<pre>\$L6: lw \$2,24(\$fp) addiu \$3,\$fp,24 addu \$2,\$3,\$2 lb \$2,4(\$2) nop slt \$2,\$2,91 beq \$2,\$0,\$L3 nop</pre>	<pre>if (a[i] <= 'Z' && a[i] >= 'A') b[i] = a[i] + 32;</pre> <p>Chuyển chữ hoa thành chữ thường.</p> <pre>#\$2 = [\$fp + 24] #\$3 = \$fp + 24 #\$2 = \$3 + \$2 #\$2 = [\$2 + 4] (\$2 chứa kí tự cần xét a[i]) #So sánh nhỏ hơn với 91 ('Z' + 1) #Nếu đúng thì đến \$L3</pre>

<pre>lw \$2,24(\$fp) addiu \$3,\$fp,24 addu \$2,\$3,\$2 lb \$2,4(\$2) nop slt \$2,\$2,65 bne \$2,\$0,\$L3 nop lw \$2,24(\$fp) addiu \$3,\$fp,24 addu \$2,\$3,\$2 lb \$2,4(\$2) nop andi \$2,\$2,0x00ff addiu \$2,\$2,32 andi \$2,\$2,0x00ff sll \$3,\$2,24 sra \$3,\$3,24 lw \$2,24(\$fp) addiu \$4,\$fp,24 addu \$2,\$4,\$2 sb \$3,1004(\$2) b \$L4 nop</pre>	<p>#Phần này lại lấy giá trị của a[i] lưu vào thanh ghi \$2</p> <p>#So sánh nhỏ hơn với 65 ('A')</p> <p>#Nếu không đúng thì đến \$L3</p> <p>#Phần này lại lấy giá trị của a[i] lưu vào thanh ghi \$2</p> <p>#Lưu giá trị của a[i] vào b[i] sau khi đã chuyển đổi.</p> <p>#\$2 = and(\$2, 0x00ff)</p> <p>#\$2 = \$2 + 32</p> <p>#\$2 = and(\$2, 0x00ff)</p> <p>#Dịch trái</p> <p>#Dịch phải => \$3 = \$2</p> <p>#Lưu vào b[i] giá trị \$3</p> <p>#Rẽ nhánh đến \$L4</p>
<pre>\$L3:</pre>	<pre>else if (a[i] <= 'z' && a[i] >= 'a') b[i] = a[i] - 32;</pre> <p>Chuyển chữ thường thành chữ hoa.</p>

lw \$2,24(\$fp)	
addiu \$3,\$fp,24	#Phần này lấy giá trị của a[i]
addu \$2,\$3,\$2	luu vào thanh ghi \$2
lb \$2,4(\$2)	
nop	
slt \$2,\$2,123	#Kiểm tra điều kiện.
beq \$2,\$0,\$L5	#So sánh \$2 nhỏ hơn 123 ('z' + 1)
nop	#Nếu đúng thì đi đến \$L5
lw \$2,24(\$fp)	
addiu \$3,\$fp,24	#Phần này lấy giá trị của a[i]
addu \$2,\$3,\$2	luu vào thanh ghi \$2
lb \$2,4(\$2)	
nop	
slt \$2,\$2,97	
bne \$2,\$0,\$L5	#So sánh \$2 nhỏ hơn 97 ('a').
nop	#Nếu sai thì đi đến \$L5
lw \$2,24(\$fp)	
addiu \$3,\$fp,24	#Phần này lấy giá trị của a[i]
addu \$2,\$3,\$2	luu vào thanh ghi \$2
lb \$2,4(\$2)	
nop	
andi \$2,\$2,0x00ff	
addiu \$2,\$2,-32	#Như đoạn code ở \$L6 đoạn code này dùng để trừ giá trị của a[i] đi 32 (chuyển thành chữ hoa) và sau đó lưu vào b[i] sử dụng hai thanh ghi trung gian là \$2 và \$3.
andi \$2,\$2,0x00ff	
sll \$3,\$2,24	
sra \$3,\$3,24	
lw \$2,24(\$fp)	

<pre> addiu \$4,\$fp,24 addu \$2,\$4,\$2 sb \$3,1004(\$2) b \$L4 nop </pre>	<pre> #Rẽ nhánh đến \$L4 </pre>
<pre> \$L5: lw \$2,24(\$fp) addiu \$3,\$fp,24 addu \$2,\$3,\$2 lb \$3,4(\$2) lw \$2,24(\$fp) addiu \$4,\$fp,24 addu \$2,\$4,\$2 sb \$3,1004(\$2) </pre>	<pre> else b[i] = a[i]; Giữ nguyên kí tự. #Nếu sai các điều kiện ở \$L6 và \$L3 sẽ đi đến đoạn code này #Phần này lấy giá trị của a[i] luu vào thanh ghi \$2 #Lấy giá trị a[i] luu vào b[i] </pre>
<pre> \$L4: lw \$2,24(\$fp) nop addiu \$2,\$2,1 sw \$2,24(\$fp) \$L2: lw \$2,24(\$fp) addiu \$3,\$fp,24 addu \$2,\$3,\$2 lb \$2,4(\$2) nop bne \$2,\$0,\$L6 </pre>	<pre> Kiểm tra điều kiện lặp của vòng while #\$2 = [\$fp + 24] #\$2 = \$2 + 1 #[\$fp + 24] = \$2 (tăng giá trị lên để chuẩn bị cho lần lặp với kí tự tiếp theo. #\$2 = [\$fp + 24] #\$3 = \$fp + 24 #\$2 = \$3 + \$2 #\$2 = [\$2 + 4] (chứa a[i] tiếp theo) #Kiểm tra điều kiện </pre>

<code>nop</code>	<code>#Nếu \$2 != 0 ('\0') thì tiếp tục lặp (đến \$L6)</code>
<code>addiu \$2,\$fp,1028</code> <code>move \$5,\$2</code> <code>lui \$2,%hi(\$LC1)</code> <code>addiu \$4,\$2,%lo(\$LC1)</code> <code>jal printf</code> <code>nop</code>	<code>printf("Chuoi sau khi da chuyen la: %s", b);</code> Hiển thị chuỗi sau khi chuyển. <code>#\$2 = \$fp + 1028</code> <code>#\$5 = \$2</code> <code>#\$4 = \$LC1</code> <code>#jal đến hàm printf để in kết quả</code>
<code>jal _getch</code> <code>nop</code>	<code>_getch();</code> Tạm dừng chương trình để xem kết quả. <code>#jal đến hàm _getch để tạm dừng chương trình, hiển thị kết quả.</code>
<code>move \$sp,\$fp</code> <code>lw \$31,2036(\$sp)</code> <code>lw \$fp,2032(\$sp)</code> <code>addiu \$sp,\$sp,2040</code> <code>j \$31</code> <code>nop</code>	Kết thúc chương trình. <code>#\$sp = \$fp</code> <code>#\$ra = [\$sp + 2036]</code> <code>#\$fp = [\$sp + 2032]</code> <code>#Dời stack point về vị trí đầu</code> <code>#Nhảy đến vị trí lưu trong \$ra lúc đầu.</code>

2. Thời gian thực thi: (Sau khi cải tiến)

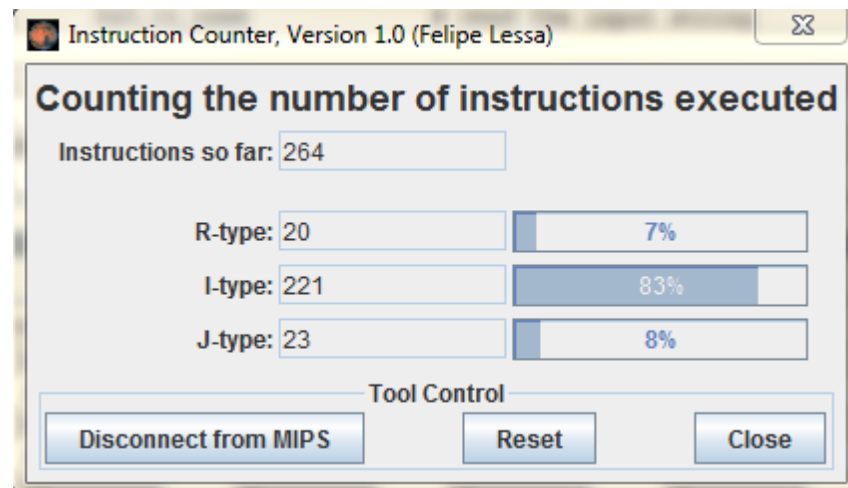
2.1. Chạy với text: “aSSIGNMENT 2 gROUP 3”

- Kết quả:

```
Nhap chuoi : aSSIGNMENT 2 gROUP 3
Chuoi sau khi chuyen : Assignment 2 Group 3
```

```
-- program is finished running --
```

- Instruction Counter:



- Thời gian thực thi:

Thời gian thực thi trên máy tính MIPS có chu kỳ T là:

$$ExeTime = CPI * IC * T = 264T$$

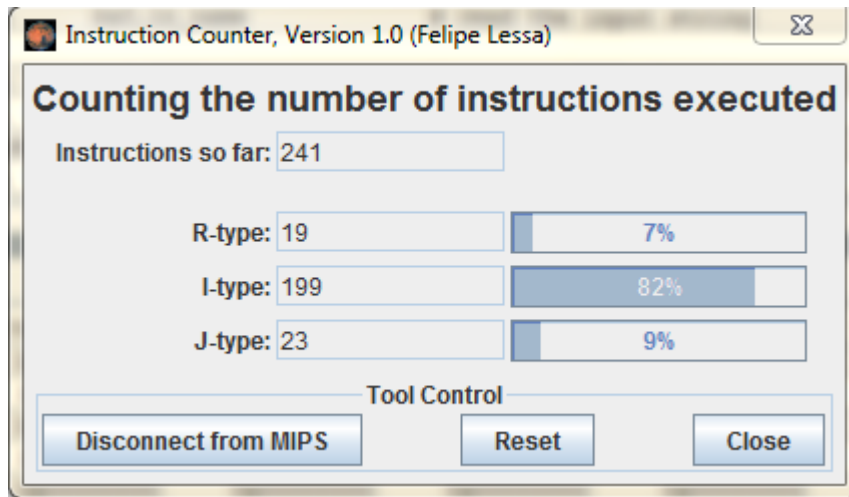
2.2. Chạy với text: “tHANK yOU sO mUCH!”

- Kết quả:

```
Nhap chuoi : tHANK yOU sO mUCH!
Chuoi sau khi chuyen : Thank You So Much!
```

```
-- program is finished running --
```

- Instruction Counter:



- Thời gian thực thi:

Thời gian thực thi trên máy tính MIPS có chu kỳ T là:

$$ExeTime = CPI * IC * T = 241T$$