ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH

BÁO CÁO MÔN HỌC

THỰC HÀNH THIẾT KẾ LUẬN LÝ SỐ

LAB 05: TRÌNH BIÊN DỊCH HỢP NGỮ

CE118.L21.2

GIẢNG VIÊN HƯỚNG DẪN: TRẦN ĐẠI DƯƠNG

SINH VIÊN THỰC HIỆN: ĐÀO CÔNG NHẬT TÂN - 19522168

TP. HỒ CHÍ MINH, 05/2021

MỤC LỤC

DANH MỤC BẢNG BIỂU

I. Trình biên dịch hợp ngữ

1. Nội dung

Viết trình biên dịch hợp ngữ bằng ngôn ngữ perl, sử dụng kiến trúc MIPS và báo cáo

2. Cách thức

-1 file .pl viết bằng perl chứa code chương trình

-1 file input.asm là file input đầu vào

-1 file input.txt là file xử lý input.asm sau khi lược bỏ chú thích và các ký tự, khoảng trắng không cần thiết

-2 file output là: DATAsegment.bin chứa mã máy trong phần .data, TEXTsegment.bin chứa mã máy trong phần .text

II. Lập trình trình biên dịch hợp ngữ

1. Xử lý file input đầu vào

Bảng 1 - Xử lý file input

code perl	Giải thích

my $fasm;
my $ftxt = 'input.txt';
my $fh;
 	
if (open($fasm, '<', $ARGV[0])) {
 	
open ($fh, '>', $ftxt);
while (my $row = <$fasm>) {
 	
chomp $row;
 	
@_str = split(/#/, $row);
$string[0] =~ s/^\s+\|\s+$//;
 	
my ($value1, $value2) = split(/ /, $string[0], 2);
my ($x, $y) = split(/ /, $value2, 2);
$value1 =~ s/\s//gs;
$value2 =~ s/\s//gs;
	Khai báo $fasm
	Khai báo $ftxt nhận đầu vào là input.txt
	Khai báo $fh
	(Hàm loại bỏ chú thích và các ký tự ko cần thiết: khoảng trắng hay tab đầu vào cuối dòng)
	Mở $fasm để đọc file đầu vào $ARGV[0]
	Mở $fh để ghi $ftxt
	Đọc từng dòng $fasm lưu vào $row
	
	Đọc $row cho đến khi xuống dòng
	
	$_str = chuỗi trước “#”
	Loại bỏ khoảng trống đầu và cuối chuỗi
	
	cắt chuỗi $string[0] lưu vào $value1 và $value2
	
	Cắt chuỗi $value2 thành 2 với //
	
	Loại bỏ khoảng trắng đầu $value1 và $value2 và $y

$y =~ s/\s//gs;
if ((length($value1) == 5) and (length($y) == 10)) {
$value1 = join(" ", $value1, $x);
$value2 = $y;
}
@_str1 = split(/,/, $value2);
$out = join(" ", @_str1);
$outer = join(" ", $value1, $out);
printf $fh "$outer\n";
}
}
close($fasm);
close($fh);
	xét đk chiều dài $value1 = 5 và $y = 10
	$value1=$value1+" "+$x
	$value2 = $y
	Loại bỏ dấu ",' thay bằng " "
	$out = chèn " "@_str1
	$outer=$value1+" "+$out
	Xuất ra file $fh biến $outer
	Đóng file $fasm
	Đóng file $fh

2. Xử lý file output DATAsegment.bin

Bảng 2 - Xử lý DATAsegment.bin

code perl	Giải thích

my $fasm;
my $fout;
open($fasm,'<', $ARGV[0]);
seek($fasm, 0, 0);
open($fout, ">DATAsegment.bin");
 	
foreach $row_i_ (<$fasm>) {
if ($row_i_ =~ /.word/){
while(scalar @DB_var % 4 != 0){
push(@DB_var, $0);
}
my @data_arr = split(/ /, $row_i_);
for (my $i = 2; $i < scalar @data_arr; $i++){
push(@DB_var, @data_arr[$i]);
push(@DB_var, $0);
push(@DB_var, $0);
push(@DB_var, $0);
}
	mở file $fasm để đọc file đầu vào
	Đặt lại nội dung của $fasm
	Mở file $fout để ghi dữ liệu từ file DATAsegment.bin
	Quét từng dòng trong $fasm
	#Hàm xét khai báo = chuỗi .word thì xử lý:
	+Số lượng phần tử trong mảng DB_var không chia hết cho 4 thì đẩy vào stack
	+cắt chuỗi $rowi với khoảng trắng
	+khởi tạo i = 2 chạy -> nhỏ hơn số lượng pt @data_arr
	+push @data_arr thứ i vào DB_var và push vào sau 3 số 0

}
elsif ($row_i_ =~ /.asciiz/){
my @data_arr = split (/[""]/, $row_i_);
my @variable = unpack("C*", $data_arr[1]);
push(@DB_var, @variable);
push(@DB_var, $0);
}
elsif ($row_i_ =~ /.ascii/){
my @data_arr = split (/[""]/, $row_i_);
my @variable = unpack("C*", $data_arr[0]);
push(@DB_var, @variable);
}
elsif ($row_i_ =~ /.byte/){
my @data_arr = split(/ /, $row_i_);
for (my $i = 2; $i < scalar @data_arr; $i++){
my $variable = $data_arr[$i];
push(@DB_var, $variable);
}
}
 	
elsif ($row_i_ =~ /.text/){
for (my $i = 0; $i < scalar @DB_var; $i++){
$DB_var[$i] = sprintf('%.2x', $DB_var[$i]);
}
 	
mv_convert_dataout();
}
}
sub mv_convert_dataout{
my $i_arr = 0;
my $row_i = 1;
while($i_arr != scalar(@DB_var)){
my $variable = join("", $DB_var[$i_arr + 3], $DB_var[$i_arr + 2],
$DB_var[$i_arr + 1], $DB_var[$i_arr]);
	#Hàm xét khai báo = chuỗi .asciiz thì xử lý:
	+Cắt chuỗi $rowi với đk “” thành @data_arr
	+Giải nén $data_arr[1] từ một chuỗi nhị phân thành một mảng
	và chuyển thành mã ascii của ký tự
	
	#Tương tự như trên
	
	#Hàm xét khai báo = chuỗi .byte thì xử lý:
	+Cắt chuỗi $rowi với đk “” thành @data_arr
	+ khởi tạo i = 2 chạy -> nhỏ hơn số lượng pt @data_arr
	+Đưa từng phần tử @data_arr vào mảng @DB_var
	#Hàm xét khai báo = chuỗi .byte thì xử lý:
	+ khởi tạo i = 2 chạy -> nhỏ hơn số lượng pt @data_arr
	+lưu chuỗi đã được format nhị phân $DB_var[$i] vào $DB_var
	+Gọi hàm mv_convert_dataout();
	#Nếu row_i gặp chuỗi có .text thì
	+chạy I = 0 , i++ đến hết mảng DB_var
	
	DB_var = DB_var[i] chuyển thành số nguyên ko dấu trong hệ hex
	
	#Hàm xử lý xuất dl ra file output
	gán i_arr = 0
	gán row_i_ = 1
	Trong khi i_arr # số lượng pt DB_var thì $variable = DBvar[$iarr+3]+””+DBvar[$iarr+2]
	+$DB_var[$i_arr + 1], $DB_var[$i_arr]);

$variable = sprintf('%08s', $variable);
$variable = substr($variable, -8);
print $fout convertbin($variable);
print $fout "\n";
$i_arr += 4;
$row_i += 1;
}
while($row_i != 1025){
print $fout "00000000\n";
$row_i += 1;
}
print $fout "\n";
 	
}
	chuyển $varibable thành chuỗi 8 ký tự ($variable ko đủ 8 thì chèn ‘ ‘ trước cho đủ 8)
	$variable = 8 ký tự cuối của nó
	Xuất ra $fout biến $variable từ hàm convertbin
	xuống dòng
	i_arr = i_arr + 4
	row_i = rowi + 1
	
	Xuất ra file $fout các số “00000000\n” với các dòng không có dư liệu tương ứng với 1024 ô nhớ
	
	xuống dòng

convertbin {
my $variable = shift;
my $len1 = length($variable);
my $len2 = $len1 * 4;
return unpack(“B$len2”, pack(“H$len1”, $variable));
}
close($fasm);
close $fout;
	Hàm chuyển hex thành bin
	Nếu $variable không phải là class
	$len1 = chiều dài chuỗi $variable
	len2 = len1 * 4
	Chuyển từng ký tự file hex thành 4 mã bin tương ứng
	
	đóng $fasm
	đóng $fout

3. Xử lý file output TEXTsegment.bin

Bảng 3 - Xử lý TEXTsegment.bin

code perl	Giải thích
#!/usr/bin/perl my %opcode = (add => '000000', addi => '001000', addiu => '001001', addu => '000000', and => '000000', andi => '001100', beq => '000100', bne => '000101', j => '000010', jal => '000011', jr => '000000', lbu => '100100', lhu => '100101', lui => '001111', lw => '100011', or => '000000', ori => '001101', slt => '000000', slti => '001010', sll => '000000', srl => '000000', sw => '101011', sub => '000000'); my %register = ('$zero' => '00000', '$at' => '00001', '$v0' => '00010', '$v1' => '00011', '$a0' => '00100', '$a1' => '00101', '$a2' => '00110', '$a3' => '00111', '$t0' => '01000', '$t1' => '01001', '$t2' => '01010', '$t3' => '01011', '$t4' => '01100', '$t5' => '01101',	**Bảng băm opcode** **Bảng băm register**

```
'$t6'   => '01110',
'$t7'   => '01111',
'$s0'   => '10000',
'$s1'   => '10001',
'$s2'   => '10010',
'$s3'   => '10011',
'$s4'   => '10100',
'$s5'   => '10101',
'$s6'   => '10110',
'$s7'   => '10111',
'$t8'   => '11000',
'$t9'   => '11001',
'$k0'   => '11010',
'$k1'   => '11011',
'$gp'   => '11100',
'$sp'   => '11101',
'$fp'   => '11110',
'$ra'   => '11111',
'$0' => '00000',
'$1' => '00001',
'$2' => '00010',
'$3' => '00011',
'$4' => '00100',
'$5' => '00101',
'$6' => '00110',
'$7' => '00111',
'$8' => '01000',
'$9' => '01001',
'$10' => '01010',
'$11' => '01011',
'$12' => '01100',
'$13' => '01101',
'$14' => '01110',
'$15' => '01111',
'$16' => '10000',
'$17' => '10001',
'$18' => '10010',
'$19' => '10011',
'$20' => '10100',
'$21' => '10101',
'$22' => '10110',
'$23' => '10111',
'$24' => '11000',
'$25' => '11001',
'$26' => '11010',
'$27' => '11011',
'$28' => '11100',
```

```
            '$29' => '11101',
            '$30' => '11110',
            '$31' => '11111');

my %format =                                    Bảng băm format
              (add   => 'R',
              addi  => 'I',
              addiu => 'I',
              addu  => 'R',
              and   => 'R',
              andi  => 'I',
              beq   => 'I1',
              bne   => 'I1',
              j     => 'J',
              jal   => 'J',
              jr    => 'RJ',
              lbu   => 'I2',
              lhu   => 'I2',
              ll    => 'I2',
              lui   => 'I3',
              lw    => 'I2',
              or    => 'R',
              ori   => 'I',
              slt   => 'R',
              slti  => 'I',
              sltiu => 'I',
              sltu  => 'I',
              sll   => 'R',
              srl   => 'R',
              sb    => 'I2',
              sc    => 'I2',
              sh    => 'I2',
              sw    => 'I2',
              sub   => 'R',
              subu  => 'R'
);
my %function = (add   => '100000',              Bảng băm function
                  addi  => '001000',
                addiu => '001001',
                addu  => '100001',
                and   => '100100',
                andi  => '001100',
                beq   => '000100',
                bne   => '000101',
                j     => '000010',
                jal   => '000011',
                jr    => '001000',
```

lbu => '100100', lhu => '100101', ll => '011110', lui => '001111', lw => '100011', nor => '011011', or => '100101', ori => '001101', slt => '101010', slti => '001010', sltiu => '001011', sltu => '101011', sll => '000000', sb => '011100', sc => '100110', sh => '011101', srl => '000010', sw => '101011', sub => '100010', subu => '010111');	
my $f1, $f2; my $dem = 0; my $opt, $opcd, $resourse, $rt, $rd, $shamt, $func;	khai báo dem = 0 khai báo các trường opcode, resourse, $ nguồn, $đích, $shamt, $function

if (open($f1, '<', $ftxt)){
open($f2, '>', 'TEXTsegment.bin');
while(my $row = <$f1>) {
chomp $row;
if ((substr($row, -2) ne ': ') and (length($row) > 1))
{$dem += 1;}
my ($out1, $out2, $out3, $out4, $out5) = split(/ /, $row, 5);
if (length($opcode{$out1}) != 6 and length($opcode{$out2}) == 6){
$out1 = $out2;
$out2 = $out3;
$out3 = $out4;
$out4 = $out5;
}
$op = $format{$out1};
$opc = $opcode{$out1};
if ($op eq 'R'){
	Mở file $f1 để đọc dl từ $ftxt
	mở $f2 để ghivàoTEXTsegment.bin
	Trong khi $row = từng dòng $f1 cho đến ký tự xuống dòng
	+Nếu 2 ký tự cuối $row ko = ':' và độ dài $row > 1 thì tăng đếm + 1
	Tách chuỗi $row thành 5 chuỗi với / / khoảng trắng
	
	+Nếu chiều dài opcode của out1 = 6 và chiều dài của opcode out2 = 6 thì gán ..
	
	$op = định dạng lệnh $out1
	$opc = opcode $out1
	+Nếu $op = R

$resourse = $register{$out3};
$rt = $register{$out4};
$rd = $register{$out2};
$shamt = '00000';
$func = $function{$out1};
my $out_res = $opc.$resourse.$rt.$rd.$shamt.$func;
 	
printf $f2 "$out_res";
}
	thì gán giá trị như bên:
	…
	….
	
	nối 5chuỗiopc,resourse,rt,rd,shamt, func lại với nhau
	
	in ra f2 $out_res (kết quả nối trên)

elsif ($op eq 'RJ'){
$resourse = $register{$out2};
my $value_ = '000000000000000';
$func = $function{$out1};
my $out_res = $opc.$resourse.$value_.$func;
printf $f2 "$out_res\n";
}
	+Nếu $op = 'RJ'
	thìbămbảng băm, gángiá trịnhưbên
	Nối 4 chuỗi opc,resourse,value,func lại với nhau
	In ra file $f2 chuỗi hoàn chỉnh sau khi nối

elsif ($op eq 'I') {
$resourse = $register{$out3};
$rt = $register{$out2};
my $binary_ = sprintf ("%.16b", $out4);
 	
$binary_ = substr($binary_, -16);
my $out_res = $opc.$resourse.$rt.$binary_;
printf $f2 "$out_res\n";
}
	+Nếu $op = I
	Gán băm bảng băm reg của out3
	$binary = chuyển $out thành số nguyên không dấu hệ bin
	$binary = chuyển $dem_ thành 16 số nhị phân không dấu
	$out_res = nối chuỗi opc,resourse,rt,binary

elsif ($op eq 'J') {
my $dem_ = 4194304;
open($fh, '<', $ftxt);
while($line1_ = <$fh>){
chomp $line1_;
my ($m1, $m2) = split(/:/, $line1_);
my ($n1, $n2) = split(/ /, $line1_);
if ($m1 eq $out2){
$out2 = $dem_;
}
if ((substr($line1_, -1) ne ':') and (length($line1_) != 0) and length($opcode{$n1}) == 6){
$dem_ += 4;
}
}
close ($fh);
 	
my $binary_ = sprintf ("%.26b", $out2);
	+Nếu $op = J
	Gán địa chỉ đếm = 0x00400000
	mở $fh đọc từ $ftxt
	*trong khi $line1_ = từng dòng $fh
	cho đến khi xuống dòng
	cắt chuỗi $line1_thành 2 phần bởi:
	cắt chuỗi $line1_thành 2 phầnbởi//
	*Nếu $m1 = $out2
	$out2 = dem
	*Nếu ký tự cuối $line1 không = ':' và độ dài $line1_ khác 0 và opcode của $n1 = 6 thì đếm = đếm + 4
	đóng file $fh
	
	$binary = chuyển $out2 thành 16 số nguyên không dấu hệ bin

$binary_ = substr($binary_, -26);
my $value_ = substr ($binary_, 24, 2);
$binary_ = substr ($binary_, 0, 24);
my $out_res = $opc.$value_.$binary_;
printf $f2 "$out_res\n";
}
elsif ($op eq 'I1'){
my $dem1 = 0;
my $dem2;
open($fh, '<', $ftxt);
while($line1_ = <$fh>){
chomp $line1_;
my ($a, $b) = split(/:/, $line1_, 2);
if ($a eq $out4){
$dem2 = $dem1;
}
if ((substr($line1_, -2) ne ': ') and (length($line1_) > 1)){
$dem1 += 1;
}
}
my $dem_ = $dem2 - $dem;
close ($fh);
my $binary_ = sprintf ("%.16b", $dem_);
$binary_ = substr($binary_, -16);
$resourse = $register{$out2};
$rt = $register{$out3};
my $out_res = $opc.$resourse.$rt.$binary_;
printf $f2 "$out_res\n";
}
elsif ($op eq 'I2'){
my ($value1, $value2) = split(/[()]/, $out3);
$resourse = $register{$value2};
$rt = $register{$out2};
my $binary_ = sprintf ("%.16b", $value1);
$binary_ = substr($binary_, -16);
my $out_res = $opc.$resourse.$rt.$binary_;
printf $f2 "$out_res\n";
}
elsif ($op eq 'I3'){
my $binary_ = sprintf ("%.16b", $out3);
my $value_ = '00000';
$rt = $register{$out2};
my $out_res = $opc.$value_.$rt.$binary_;
printf $f2 "$out_res\n";
}
	Lấy 26 ký tự cuối của $binary
	lấy 2 ký tự tính từ ký tự thứ 24 của chuỗi $binary
	Nối chuỗi opc,value,binary thành out_res rồi in ra file $f2
	
	+Nếu $op = I1
	dem1 = 0
	…
	Mở $fh để đọc từ file ftxt
	*Trong khi $line1 = từng dòng của $fh cho đến khi xuống dòng
	Cắt chuỗi $line1_ thành 2 với ‘:’
	*Nếu $a = $out4
	dem2 = dem1
	
	Nếu 2 ký tự cuối $line1_ # ‘:’ và chiều dài chuỗi $line1_ > 1 thì đêm ++
	..
	$dem_ = địa chỉ tương đối dem2 – dem1
	$binary = chuyển $dem_ thành 16 số nhị phân không dấu
	resourse = regisster của out2
	rt = register của out3
	nối chuỗi $out_res với 4 chuỗi opc,resourse,rt,binary sau đó in vào $f2
	+Nếu $op = I2
	Cắt chuỗi $out3 thành 2 với ()
	resourse = register của value2
	rt = register của out2
	$binary = chuyển $value1 thành 16 số nhị phân không dấu
	$binary = 16 ký tự cuối chuỗi của nó
	Nối chuỗi thành out_res 2 sau đó in vào file $f2
	+Nếu $op = I3
	$binary = chuyển $out3 thành 16 số nguyên không dấu hệ bin
	value = 00000
	rt = register của out2
	nối 4 chuỗi opc, value,rt,binary thành out_res rồi in vào file $f2

} }	
close($f1); close($f2);	Đóng file $f1 Đóng file $f2