ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ Thông tin & Truyền thông

A picture containing icon

Description automatically generated

IT3280

Lý thuyết và ngôn ngữ hướng đối tượng

BÁO CÁO PROJECT

CUỐI KỲ

Bài 6

Hàm cấp phát bộ nhớ Malloc()

Họ và tên: Nguyễn Thành Dương

MSSV: 20194531

# Giới thiệu bài toán

Cho chương trình hàm malloc(), kèm theo đó là ví dụ minh họa, được viết bằng hợp ngữ MIPS, để cấp phát bộ nhớ cho một biến con trỏ nào đó. Hãy đọc chương trình và hiểu rõ nguyên tắc cấp phát bộ nhớ động.Trên cơ sởđó, hãy hoàn thiện chương trình như sau. Lưu ý, ngoài viết các hàm đó, cần viết thêm một số ví dụ minh họa để thấy việc sử dụng hàm đó như thế nào.

1)Việc cấp phát bộ nhớ kiểu word/mảng wordcó 1 lỗi, đó là chưa bảo đảm qui tắc địa chỉcủa kiểu wordphải chia hết cho 4. Hãy khắc phục lỗi này.

2)Viết hàm lấy giá trịWord /Byte của biến con trỏ(tương tựnhư \*CharPtr, \*BytePtr, \*WordPtr)

3)Viết hàm lấy địa chỉ biến con trỏ (tương tựnhư &CharPtr, &BytePtr, \*WordPtr)

4)Viết hàm thực hiện copy 2 con trỏ xâu kí tự(Xem ví dụvềCharPtr)

5)Viết hàm tính toàn bộ lượng bộ nhớ đã cấp phát cho các biến động

6)Hãy viết hàm Malloc2 để cấp phát cho mảng 2 chiều kiểu .word với tham số vào gồm:

a.Địa chỉ đầu của mảng

b.Số dòng

c.Số cột

7)Tiếp theo câu 6, hãy viết 2 hàm GetArray[i][j] và SetArray[i][j] để lấy/thiết lập giá trị cho phần tử ở dòng I cột j của mảng.

# 2.Phương pháp thực hiện

1)Khi tính toán bộ nhớ cần cấp phát thì chúng ta chia nó cho 4 nếu có số dư thì thêm 1 ô nhớ và tính toán lại.

2)Biến con trỏ sẽ lưu giá trị đầu tiên của bộ nhớ được cấp phát từ đó chúng ta sẽ lấy các giá trị ra.

3)Biến con trỏ sẽ được lấy ra như bình thường.

4)Khi copy 2 con trỏ sâu thì:

-Đầu tiên chúng ta cấp phát bộ nhớ mới bằng với bộ nhớ của cn trỏ cần copy

-Sau đó lấy địa chỉ con trỏ dùng vòng lặp để copy với số lần là số ô nhớ.

5) Sys\_MyFreeSpace: dùng để cấp bộ nhớ cho các biến con trỏ.

Sys\_TheTopOfFree: biến chứ địa chỉ đầu tiên của vùng nhớ còn trống

Lấp hiệu 2 biến sẽ ra bộ nhớ đã cấp phát cho biến động

6)-Mảng 2 chiều bản chất giống với cấp phát mảng 1 chiều

-Mảng 2 chiều chỉ khác ở cách tính ô nhớ và cách truy cập ô nhớ

- Cách tính tính bộ nhớ là: Số hàng X Số cột X Kích thước 1 ô

7)Khi truy cập vùng nhớ mảng 2 chiều tính theo công thức:

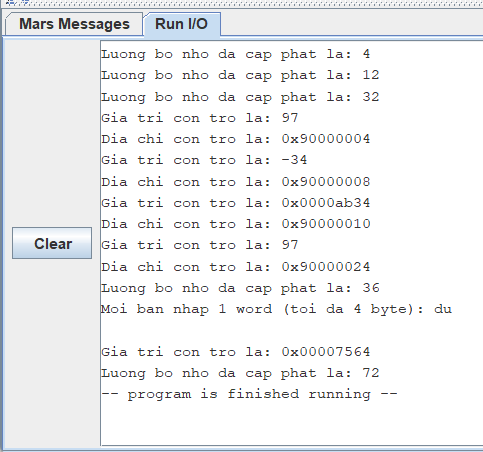
Số cột X ( Số hàng – 1 ) + ( Số cột – 1 )

# 3.Kết quả

Đã làm và thực hiện hết yêu cầu đề ra

Graphical user interface, text, application

Description automatically generated



# 4.Mã nguồn

#================================ Computer Architecture ============================

# @Author : Nguyen Thanh Duong

# @StudentID: 20194531

# @Language : Assembly, MIPS

#===================================================================================

.data

CharPtr: .word 0 # Bien con tro, tro toi kieu asciiz

BytePtr: .word 0 # Bien con tro, tro toi kieu Byte

WordPtr: .word 0 # Bien con tro, tro toi mang kieu Word

CharPtrCopy: .word 0 # Bien con tro, tro toi mang Copy kieu Word

WordPtr2: .word 0 # Bien con tro, tro toi mang 2 chieu kieu Word

Enter: .word 1 # Buffer store word

ValueResult: .asciiz "\nGia tri con tro la: "

AddressResult: .asciiz "\nDia chi con tro la: "

MemoryResult: .asciiz "\nLuong bo nho da cap phat la: "

RowMessage: .asciiz "Moi ban nhap so dong"

ColMessage: .asciiz "Moi ban nhap so cot"

Notification: .asciiz "Phuong thuc GetArray va SetArray !!!"

Message: .asciiz "Nhap 0 de thiet lap gia tri\nNhap 1 de lay gia tri\nNhap 2 de thoat ham"

EnterMessage: .asciiz "\nMoi ban nhap 1 word (toi da 4 byte): "

.kdata

# Bien chua dia chi dau tien cua vung nho con trong

Sys\_TheTopOfFree: .word 1

# Vung khong gian tu do, dung de cap bo nho cho cac bien con tro

Sys\_MyFreeSpace:

.text

#Khoi taovung nho cap phat dong

jal SysInitMem

#-----------------------

# Tinh gia tri bo nho da cap phat

#-----------------------

jal memoryAllocated

nop

#-----------------------

# Cap phat cho bien con tro, gom 3 phan tu, moi phan tu 1 byte

# $s5: Luu so thanh ghi can de cap phat CharPtr

#-----------------------

la $a0, CharPtr

addi $a1, $zero, 3

addi $a2, $zero, 1

jal malloc

add $s5, $0, $t1

#-----------------------

# Tinh gia tri bo nho da cap phat

#-----------------------

jal memoryAllocated

nop

#-----------------------

# Cap phat cho bien con tro, gom 6 phan tu, moi phan tu 1 byte

# $s6: Luu so thanh ghi can de cap phat BytePtr

#-----------------------

la $a0, BytePtr

addi $a1, $zero, 6

addi $a2, $zero, 1

jal malloc

add $s6, $0, $t1

#-----------------------

# Tinh gia tri bo nho da cap phat

#-----------------------

jal memoryAllocated

nop

#-----------------------

# Cap phat cho bien con tro, gom 5 phan tu, moi phan tu 4 byte

# $s7: Luu so thanh ghi can de cap phat WordPtr

#-----------------------

la $a0, WordPtr

addi $a1, $zero, 5

addi $a2, $zero, 4

jal malloc

add $s7, $0, $t1

#-----------------------

# Tinh gia tri bo nho da cap phat

#-----------------------

jal memoryAllocated

nop

#-----------------------

# Gan gia tri cho con tro

#-----------------------

la $a0, CharPtr

lw $t1, 0($a0)

addi $t0, $0, 'a'

sb $t0, 0($t1)

addi $t0, $0, 'b'

sb $t0, 1($t1)

addi $t0, $0, 'c'

sb $t0, 2($t1)

la $a0, BytePtr

lw $t1, 0($a0)

addi $t0, $0, -34

sb $t0, 0($t1)

la $a0, WordPtr

lw $t1, 0($a0)

addi $t0, $0, 0xab34

sw $t0, 0($t1)

#-----------------------

# Lay gia tri va dia chi cua Word/Byte

# $a0: Dia chi con tro

# $t1: Dia chi vung nho duoc cap phat ( dia chi con tro )

# $t0: Gia tri bien con tro

#-----------------------

la $a0, CharPtr

lw $t1, 0($a0)

lb $t0, 0($t1)

jal getValue

nop

jal getAddress

la $a0, BytePtr

lw $t1, 0($a0)

lb $t0, 0($t1)

jal getValue

nop

jal getAddress

la $a0, WordPtr

lw $t1, 0($a0) # Lay ra dia chi cua con tro

lw $t0, 0($t1) # Lay ra gia tri word tai dia chi con tro

jal getValue\_word

nop

jal getAddress

#-----------------------

# Copy con tro xau ki tu (CharPtr)

#-----------------------

la $a0, CharPtrCopy

la $a1, CharPtr

add $s4, $0, $s5 # Lay kich thuoc mang copy

jal CopyChar

# Kiem tra ket qua Copy

la $a0, CharPtrCopy

lw $t1, 0($a0)

lb $t0, 0($t1)

jal getValue

nop

jal getAddress

nop

#-----------------------

# Tinh gia tri bo nho da cap phat

#-----------------------

jal memoryAllocated

nop

#-----------------------

# Cap phat mang 2 chieu kieu .word voi tham so dau vao

# $s3 = i: so dong

# $s2 = j: so cot

#-----------------------

SetRow: li $v0, 51

la $a0, RowMessage

syscall

slt $s3, $a0, $0 # Kiem tra gia tri nhap vao phai lon hon 0

bne $s3, $0, SetRow

add $s3, $0, $a0

SetCol: li $v0, 51

la $a0, ColMessage

syscall

slt $s2, $a0, $0 # Kiem tra gia tri nhap vao phai lon hon 0

bne $s2, $0, SetCol

add $s2, $0, $a0

la $a0, WordPtr2 # Load dia chi cua mang 2 chieu

mul $a1, $s3, $s2 # Kich thuoc mang 2 chieu

addi $a2, $zero, 4 # Kich thuoc phan tu mang

jal malloc2

nop

#-----------------------

# SetArray[i][j] && GetArray[i][j]

# $t0 = i: so dong < $s3

# $t1 = j: so cot < $s2

#-----------------------

Notifi: li $v0, 55

la $a0, Notification # In thong bao chuyen sang ham Get va Set

li $a1, 1

syscall

EnterRow: li $v0, 51

la $a0, RowMessage

syscall

sle $t0, $a0, $s3 # Kiem tra so dong nhap vao nho hon so dong cap phat

beq $t0, $0, EnterRow

beq $a0, $0, EnterRow # So dong nhap vao lon hon 0

add $t0, $0, $a0

EnterCol: li $v0, 51

la $a0, ColMessage

syscall

sle $t1, $a0, $s2 # Kiem tra so cot nhap vao nho hon so cot cap nhat

beq $t1, $0, EnterCol

beq $a0, $0, EnterCol # So cot nhap vao lon hon 0

add $t1, $0, $a0

addi $t2, $0, 4 # Kich thuoc moi phan tu 4 byte

li $v0, 51 # Lua chon Get hoac Set hoac thoat chuong trinh

la $a0, Message

syscall

add $a1, $0, $a0

la $a0, WordPtr2

beq $a1, $0, SetArray

nop

beq $a1, 1, GetArray

nop

#-----------------------

# Tinh gia tri bo nho da cap phat

#-----------------------

jal memoryAllocated

nop

lock: li $v0, 10

syscall

nop

# ==================================== END MAIN =====================================

#------------------------------------------

# Ham khoi tao cho viec cap phat dong

# @param khong co

# @detail Danh dau vi tri bat dau cua vung nho co the cap phat duoc

#------------------------------------------

SysInitMem:

la $t9, Sys\_TheTopOfFree # Lay con tro chua dau tien con trong, khoi tao

la $t7, Sys\_MyFreeSpace # Lay dia chi dau tien con trong, khoi tao

sw $t7, 0($t9) # Luu lai

jr $ra

#------------------------------------------

# Ham cap phat bo nho dong cho cac bien con tro

# @param [in/out] $a0 Chua dia chi cua bien con tro can cap phat

# Khi ham ket thuc, dia chi vung nho duoc cap phat se luu tru vao bien con tro

# @param [in] $a1 So phan tu can cap phat

# @param [in] $a2 Kich thuoc 1 phan tu, tinh theo byte

# @return $v0 Dia chi vung nho duoc cap phat

#------------------------------------------

malloc: la $t9, Sys\_TheTopOfFree # Lay con tro chua dau tien con trong, khoi tao

lw $t8, 0($t9) # Lay dia chi dau tien con trong

sw $t8, 0($a0) # Cat dia chi do vao bien con tro

addi $v0, $t8, 0 # Dong thoi la ket qua tra ve cua ham

mul $t7, $a1, $a2 # Tinh kich thuoc cua mang can cap phat = so phan tu \* kich thuoc phan tu

addi $t0, $0, 4 # Luu kich thuoc kieu word de tinh so luong thanh ghi can cap

div $t7, $t0 # So luong thanh ghi can cap phat

mflo $t1 # $lo luu thuong

mfhi $t2 # $hi luu so du

beq $t2, $0, allocation

addi $t1, $t1, 1 # Cap them 1 thanh ghi de luu

# Cap phat bo nho chia het 4 cho cac bien con tro

allocation:

mul $t7, $t1, $t0 # Kich thuoc mang can cap phat

add $t6, $t8, $t7 # Tinh dia chi dau tien con trong

sw $t6, 0($t9) # Luu tro lai dia chi dau tien do vao bien Sys\_TheTopOfFree

jr $ra

#------------------------------------------

# Ham lay gia tri cua bien con tro (\*CharPtr, \*BytePtr, \*WordPtr)

# getValue: dung de tra gia tri cua \*CharPtr, \*BytePtr

# getValue\_word: dung de tra gia tri cua \*WordPtr

#------------------------------------------

getValue:

li $v0, 4 # In message

la $a0, ValueResult

syscall

li $v0, 1

add $a0, $0, $t0

syscall

jr $ra

getValue\_word:

li $v0, 4 # In message

la $a0, ValueResult

syscall

li $v0, 34

add $a0, $0, $t0

syscall

jr $ra

#------------------------------------------

# Ham lay dia chi cua bien con tro (&CharPtr, &BytePtr, &WordPtr)

# @param [in] $t1: Dia chi vung nho duoc cap phat ( dia chi con tro )

#------------------------------------------

getAddress: li $v0, 4 # In message

la $a0, AddressResult

syscall

li $v0, 34 # In dia chi con tro

add $a0, $0, $t1

syscall

jr $ra

#------------------------------------------

# Ham copy con tro xau ki tu

# @param [in/out] $a0 Chua dia chi cua bien con tro can cap phat

# Khi ham ket thuc, dia chi vung nho duoc cap phat se luu tru vao bien con tro

# $a1 Chua dia chi cua bien con tro muon sao chep

# @param [in] $s4 So thanh ghi can cap phat

# return $v0 Dia chi vung nho duoc cap phat

#------------------------------------------

CopyChar: la $t9, Sys\_TheTopOfFree # Lay con tro chua dau tien con trong, khoi tao

lw $t8, 0($t9) # Lay dia chi dau tien con trong

sw $t8, 0($a0) # Cat dia chi do vao bien con tro

add $v0, $0, $t8

mul $t7, $s4, 4 # Kich thuoc bo nho can cap phat

add $t6, $t8, $t7 # Tinh dia chi dau tien con trong

sw $t6, 0($t9) # Luu tro lai dia chi con trong do vao Sys\_TheTopOfFree

add $t5, $0, $s4 # Luu so thanh ghi can load de copy

lw $t1, 0($a1) # Load dia chi cua con tro can Copy

lw $t0, 0($a0) # Load dia chi con tro Copy

Copy: lw $t4, 0($t1) # Load du lieu cua con tro can Copy

sw $t4, 0($t0) # Store du lieu vao con tro Copy

addi $t1, $t1, 4 # Tang dia chi con tro can Copy

addi $t0, $t0, 4 # Tang dia chi con tro Copy3

addi $t5, $t5, -1 # Giam so luong thanh ghi can Copy

bne $t5, $0, Copy # Neu so thanh ghi can load khac 0 thi tiep tuc Copy

nop

jr $ra

# Copy done

#------------------------------------------

# Ham lay luong bo nho da cap phat

# Method : Lay Dia chi o Sys\_TheTopOfFree - Sys\_MyFreeSpace

#------------------------------------------

memoryAllocated:

li $v0, 4 # In message

la $a0, MemoryResult

syscall

li $v0, 1 # Tinh dung luong bo nho

la $a1, Sys\_TheTopOfFree

lw $a1, 0($a1)

la $a2, Sys\_MyFreeSpace

sub $a0, $a1, $a2

syscall

jr $ra

#------------------------------------------

# Ham cap phat bo nho dong cho mang 2 chieu kieu word

# @param [in/out] $a0 Chua dia chi cua bien con tro can cap phat

# Khi ham ket thuc, dia chi vung nho duoc cap phat se luu tru vao bien con tro

# @param [in] $a1 So phan tu can cap phat

# @param [in] $a2 Kich thuoc 1 phan tu, tinh theo byte

#------------------------------------------

malloc2: la $t9, Sys\_TheTopOfFree # Lay con tro chua dau tien con trong, khoi tao

lw $t8, 0($t9) # Lay dia chi dau tien con trong

sw $t8, 0($a0) # Cat dia chi do vao bien con tro

mul $t7, $a1, $a2 # Kich thuoc mang can cap phat

add $t8, $t8, $t7 # Tinh dia chi dau tien con trong

sw $t8, 0($t9) # Luu tro lai dia chi dau tien do vao bien Sys\_TheTopOfFree

jr $ra

#------------------------------------------

# Ham thiet lap gia tri cho phan tu o dong i cot j cua mang 2 chieu kieu word

# @param [in] $a0 Dia chi mang 2 chieu

# @param [in] $s2 So cot cua mang 2 chieu

# @param [in] $t0 Vi tri hang cua mang 2 chieu

# @param [in] $t1 Vi tri cot cua mang 2 chieu

# @param [in] $t2 Kich thuoc 1 phan tu word

# Khoang cua phan tu = (hang - 1)\*Socot + cot-1

#------------------------------------------

SetArray: lw $t3, 0($a0)

addi $t0, $t0, -1 # Hang - 1

mul $t0, $t0, $s2 # (Hang-1)\*SoCot

addi $t1, $t1, -1 # cot-1

add $t0, $t0, $t1 # (Hang-1)\*SoCot + cot-1

mul $t0, $t0, $t2 # Khoang cach tu con tro dau tien den phan tu can thay doi

# 4 \* ((Hang-1)\*SoCot + cot-1)

add $t3, $t3, $t0 # Dia chi cua phan tu

li $v0, 4 # In message bat dau nhap

la $a0, EnterMessage

syscall

add $s0, $0, $0 # i = 0

la $s1, Enter

ReadChar:

li $v0, 12 # Read char

syscall

CheckChar:

beq $v0, 10, Return # Kiem tra ky tu enter

add $t4, $s1, $s0 # $t4 = dia chi cua string[i] nhap vao

sb $v0, 0($t4) # dua ki tu vao string nhap

add $t5, $t3, $s0 # $t5 = dia chi cua WordPtr2[i][j][k] k < 4

sb $v0, 0($t5)

addi $s0, $s0, 1 # i = i + 1

slti $t5, $s0, 4 # if i < 4

beq $t5, $0, Return

nop

j ReadChar

nop

Return: j EnterRow

#------------------------------------------

# Ham lay gia tri cho phan tu o dong i cot j cua mang 2 chieu kieu word

# @param [in] $a0 Dia chi mang 2 chieu

# $s2 So cot cua mang 2 chieu

# $s3 So hang cua mang 2 chieu

# $t0 Vi tri hang cua mang 2 chieu

# $t1 Vi tri cot cua mang 2 chieu

# $t2 Kich thuoc 1 phan tu word

# Khoang cua phan tu = (hang - 1)\*Socot + cotword-1

#------------------------------------------

GetArray: lw $t3, 0($a0) # Lay dia chi phan tu dau tien cua mang

addi $t0, $t0, -1 # Hang - 1

mul $t0, $t0, $s2 # (Hang-1)\*SoCot

addi $t1, $t1, -1 # Cot-1

add $t0, $t0, $t1 # (Hang-1)\*SoCot + cot-1

mul $t0, $t0, $t2 # Khoang cach tu con tro dau tien den phan tu can thay doi

# 4 \* ((Hang-1)\*SoCot + cot-1 )

add $t3, $t3, $t0 # Dia chi cua phan tu

li $v0, 4 # In message

la $a0, ValueResult

syscall

li $v0, 34 # In gia tri ra

lw $a0, 0($t3) # Lay ra gia tri cua phan tu tai hang i, cot j

syscall

j EnterRow