Báo cáo mini project

# Đề bài

(8). Write a program to:

You have to create 1 variable (number students) and 2 array (student name, mark)

to store the input data

- Input the number of students in class.

- Input the name of students in class, Math mark (0 →10)

- Sort students due to their mark.

# Ý tưởng giải quyết

* Lưu trữ các record cho từng sinh viên (8 byte):
  + Tên sv: địa chỉ kí tự đầu của xâu kí tự bị kết thúc bởi null (4 byte)
  + Điểm số: giá trị word (4 byte)
* Lưu tối đa 100 records => cần reserve 800 byte (recordArr)
* Lưu trữ xâu kí tự:
  + Sử dụng reserved space 10000 byte (strPool) => mỗi xâu tên của record có trung bình 100 byte
  + Khi tạo xâu mới, bắt đầu ghi tại địa chỉ đánh dấu bởi strTop
* Sắp xếp: thuật toán nổi bọt
  + Dễ cài đặt
  + Sử dụng ít biến cục bộ

# Code

.data

strPool: .space 10000

strTop: .word 0 # address of top of strPool

studentNum: .word 0

recordArr: .space 800 # space for 100 records, 8 bytes for each record

msgAskStudentNum: .asciiz "Number of students in class: "

msgName: .asciiz "Name: "

msgScore: .asciiz "Score: "

msgPrintStudents: .asciiz "\n=====Students=====\n"

msgSort: .asciiz "\nSorting students ...\n"

.text

main: la $t0, strPool

sw $t0, strTop

# ask for number of students in class

addi $v0, $0, 4

la $a0, msgAskStudentNum

syscall

addi $v0, $0, 5

syscall

sw $v0, studentNum

# user input loop

addi $t0, $0, 0

lw $t1, studentNum

main\_input\_loop:

beq $t0, $t1, main\_done\_input

nop

sll $t4, $t0, 3

la $t3, recordArr

addu $t3, $t3, $t4 # $t3: base address of current record

# ask student's name

addi $v0, $0, 4

la $a0, msgName

syscall

addi $v0, $0, 8

lw $a0, strTop

addi $a1, $0, 100

syscall

jal strLen

nop

# save student's name

sw $a0, 0($t3) # save name to record

addi $t4, $v0, 0

addu $t4, $t4, $a0

sb $0, 0($t4) # add null character

addi $t4, $t4, 1

sw $t4, strTop

# ask student's score

addi $v0, $0, 4

la $a0, msgScore

syscall

addi $v0, $0, 5

syscall

# save student's score

sw $v0, 4($t3)

addi $t0, $t0, 1

j main\_input\_loop

nop

main\_done\_input:

jal sortStudents

nop

jal printStudents

nop

# Exitting syscall

addi $s0, $v0, 0

addi $v0, $zero, 10

syscall

# strlen()

# $a0: stringBuffer

# $v0: output length

strLen: sw $ra, -4($sp)

sw $fp, -8($sp)

jal prepareStack

nop

addi $v0, $0, 0

addi $t0, $a0, 0

strLen\_loop:

lb $t1, 0($t0)

beq $t1, $0, strLen\_done\_loop

nop

addi $t0, $t0, 1

addi $v0, $v0, 1

j strLen\_loop

nop

strLen\_done\_loop:

jal destroyStack

nop

lw $ra, -4($sp)

lw $fp, -8($sp)

jr $ra

nop

# void printStudents()

printStudents:

sw $ra, -4($sp)

sw $fp, -8($sp)

jal prepareStack

nop

addi $v0, $0, 4

la $a0, msgPrintStudents

syscall

addi $t0, $0, 0

lw $t1, studentNum

printStudents\_loop:

beq $t0, $t1, printStudents\_done\_loop

nop

sll $t2, $t0, 3

la $t3, recordArr

addu $t3, $t3, $t2 # $t3: base address of current record

# print name

addi $v0, $0, 4

la $a0, msgName

syscall

lw $a0, 0($t3)

syscall

# print score

la $a0, msgScore

syscall

addi $v0, $0, 1

addi $t4, $t3, 4

lw $a0, 0($t4)

syscall

addi $v0, $0, 11

li $a0, '\n'

syscall

# increment $t0

addi $t0, $t0, 1

j printStudents\_loop

nop

printStudents\_done\_loop:

jal destroyStack

nop

lw $ra, -4($sp)

lw $fp, -8($sp)

jr $ra

nop

# void sortStudents()

sortStudents:

sw $ra, -4($sp)

sw $fp, -8($sp)

jal prepareStack

nop

addi $v0, $0, 4

la $a0, msgSort

syscall

lw $t0, studentNum

addi $t1, $0, 0 # i

addi $t2, $t0, -1

sortStudents\_loop1:

beq $t1, $t2, sortStudents\_done\_loop1

nop

addi $t3, $0, 0 # swapped = false

addi $t4, $0, 0 # j

sub $t5, $t2, $t1

sortStudents\_loop11:

beq $t4, $t5, sortStudents\_done\_loop11

nop

sll $t7, $t4, 3

la $t6, recordArr

addu $t6, $t6, $t7 # $t6: base address of j\_th record

lw $t7, 4($t6)

lw $t8, 12($t6)

ble $t7, $t8, sortStudents\_skip\_swap

nop

sw $t7, 12($t6)

sw $t8, 4($t6)

lw $t7, 0($t6)

lw $t8, 8($t6)

sw $t7, 8($t6)

sw $t8, 0($t6)

addi $t3, $0, 1 # swapped = true

sortStudents\_skip\_swap:

addi $t4, $t4, 1

j sortStudents\_loop11

nop

sortStudents\_done\_loop11:

beq $t3, $0, sortStudents\_done\_loop1

addi $t1, $t1, 1

j sortStudents\_loop1

nop

sortStudents\_done\_loop1:

jal destroyStack

nop

lw $ra, -4($sp)

lw $fp, -8($sp)

jr $ra

nop

prepareStack:

sw $t0, -12($sp)

sw $t1, -16($sp)

sw $t2, -20($sp)

sw $t3, -24($sp)

sw $t4, -28($sp)

sw $t5, -32($sp)

sw $t6, -36($sp)

sw $t7, -40($sp)

sw $t8, -44($sp)

sw $t9, -48($sp)

addi $sp, $sp, -48

jr $ra

nop

destroyStack:

addi $sp, $sp, 48

lw $t0, -12($sp)

lw $t1, -16($sp)

lw $t2, -20($sp)

lw $t3, -24($sp)

lw $t4, -28($sp)

lw $t5, -32($sp)

lw $t6, -36($sp)

lw $t7, -40($sp)

lw $t8, -44($sp)

lw $t9, -48($sp)

jr $ra

nop

# Kết quả

Graphical user interface, application

Description automatically generated