

Evaluación del Trabajo de Laboratorio Número 1

Arquitectura de Computadoras

21/04/2022

Instrucciones

- La presente evaluación es individual y podrá utilizar todo el material escrito del que disponga.
- La misma consta de **2** ejercicios en total y la duración será de **90** minutos.
- En cada ejercicio se indica el tiempo estimado para resolverlo y el puntaje que otorga el mismo.
- La evaluación se desarrollará en el entorno de Google Classroom.
- La evaluación deberá ser desarrollada en el documento de Google Docs adjunto al enunciado.
- Se deberá completar el encabezado del documento con los datos del alumno.
- No se admitirán otros archivos o formatos a menos que sean explícitamente solicitados.
- No se permitirá intercambiar ni prestar ningún tipo de elementos o información entre los estudiantes.
- Para hacer sus soluciones se recomienda enfáticamente que pruebe sus códigos en el simulador Venus (<https://venus.cs61c.org>) antes de enviarlos, ya que si no funciona en el simulador el puntaje se verá significativamente disminuido.
- La legibilidad del código, la cantidad de comentarios y la buena presentación serán tenidas en cuenta para la calificación final.

Ejercicios para resolver

1) 90 Minutos - 100 Puntos (15,20,30,10,25)

Se desea escribir un programa en lenguaje ensamblador RV32I que convierta una cadena de caracteres que contiene el nombre completo de una persona para que sólo la primera letra del nombre esté en mayúsculas y el apellido completamente en mayúsculas (por ejemplo "nOmBrE ApELLiDo" lo convierte a "Nombre APELLIDO"). Se sabe que todas las cadenas a procesar tienen solamente un nombre y un apellido.

- a) Escriba una subrutina **cambiar** que cambie un carácter en memoria a mayúscula o minúscula según como sea invocada. La misma recibirá la dirección del carácter en memoria, un límite inferior (la letra **a** o **A** según corresponda), un límite superior (la letra **z** o **Z**) y una cantidad a sumar (o restar) al carácter recibido si el mismo está dentro de los límites. Esta subrutina recibe todos los parámetros por registros y retorna en el registro **a0** una copia del carácter después de la conversión.
- b) Escriba una subrutina **apellido** que, utilizando la subrutina anterior, convierta las letras de una palabra a mayúsculas. Esta subrutina recibe el puntero al primer carácter de la palabra a cambiar y debe terminar su trabajo cuando encuentra un espacio o termina la cadena. Si lo considera necesario esta subrutina puede devolver un valor.
- c) Escriba una subrutina **nombre**, que utilizando la subrutina **cambiar**, convierta la primera letra de una palabra a mayúsculas y el resto a minúsculas. Esta subrutina recibe el puntero al primer carácter de la palabra a cambiar y debe terminar su trabajo cuando encuentra un espacio o termina la cadena. Si lo considera necesario esta subrutina puede devolver un valor.
- d) Escriba una subrutina **mostrar** que reciba la dirección de memoria del primer carácter de una cadena y la muestre por pantalla.
- e) Utilizando las subrutinas anteriores, escriba un programa que convierta una cadena según el enunciado inicial del problema y muestre por pantalla la cadena antes y después de la conversión.

Soluciones completas

1)

```

.data
prueba: .asciiz "n0mBrE ApEllido"

.text
.globl main

main:
    la s0, prueba          # Se carga el puntero a la cadena
    mv a0,s0               # Se define el parametro para la subrutina
    jal mostrar_cadena     # Se muestra la cadena inicial

    mv a0,s0               # Se define el parametro para la subrutina
    jal nombre             # Se convierte el nombre
    addi a0,a0,1           # Se omite el espacio entre nombre y apellido
    jal apellido           # Se convierte el apellido

    li a0,11               # Se define el parametro para imprimir un caracter
    li a1,'\n'             # Se define el caracter como retorno de carro
    ecall                  # Se muestra un retorno de carro en pantalla

    mv a0,s0               # Se define el parametro para la subrutina
    jal mostrar_cadena     # Se muestra la cadena convertida

    li a0, 10              # Servicio del entorno para terminar
    ecall

cambiar:                  # Función para convertir un caracter
    lb a0,0(a1)            # Cargo el caracter con la letra a comparar
    blt a0,a2,fin_ca       # Terminar si el caracter es menor al inferior
    bgt a0,a3,fin_ca       # Terminar si el caracter no esta en rango
    add a0,a0,a4            # Cambiar el caracter si esta en el rango
    sb a0,0(a1)            # Actualizar el caracter cambiado
fin_ca: ret                # Terminar la funcion y retornar

apellido:                 # Funcion para cambiar el formato a los apellidos
    addi sp,sp,-8          # Reservar espacio para un registro en la pila
    sw ra,0(sp)            # Conservar el valor de la direccion de retorno
    sw s0,4(sp)            # Conservar el valor del registro salvado s0
    mv a1, a0              # Se mueve la direccion inicial de la cadena
    li a2,'a'              # Limite inferior de los caracteres a convertir
    li a3,'z'              # Limite superior de los caracteres a convertir
    li a4,-32              # Decrementar 32 para convertir en mayusculas
    li s0,' '              # Se carga el caracter de espacio

lzo_ap:
    jal cambiar            # Llamar a la subrutina para convertir el caracter
    beqz a0,fin_ap         # Terminar si el caracter es el terminador nulo
    beq a0,s0,fin_ap       # Comparar el caracter con el espacio
    addi a1,a1,1           # Incrementar en un el puntero al caracter
    j lzo_ap               # Repetir el lazo para el siguiente caracter

fin_ap:
    lw s0,4(sp)            # Recuperar el valor de los registros de parametros
    lw ra,0(sp)            # Recuperar el valor de la direccion de retorno
    addi sp,sp,8           # Devolver el espacio para cinco registros a la pila
    mv a0,a1               # Recupera el valor del puntero
    ret                    # Terminar la funcion y retornar

nombre:
    addi sp,sp,-8          # Función para cambiar el formato a los nombres
    sw ra,0(sp)            # Reservar espacio para un registro en la pila
    sw s0,4(sp)            # Conservar el valor de la direccion de retorno
    # Conservar el valor del registro salvado s0

```

```

mv a1, a0          # Se mueve la direccion inicial de la cadena
li a2, 'a'         # Limite inferior de los caracteres a convertir
li a3, 'z'         # Limite superior de los caracteres a convertir
li a4, -32         # Decrementar 32 para convertir en mayusculas
li s0, ' '         # Se carga el caracter de espacio

jal cambiar        # Llamar a la subrutina para convertir el caracter
li a2, 'A'         # Limite inferior de los caracteres a convertir
li a3, 'Z'         # Limite superior de los caracteres a convertir
li a4, 32          # Incrementar 32 para convertir en minusculas

lzo_no:
beqz a0, fin_no    # Terminar si el caracter es el terminador nulo
beq a0, s0, fin_no # Comparar el caracter con el espacio
addi a1, a1, 1     # Incrementar en un el puntero al caracter
jal cambiar        # Llamar a la subrutina para convertir el caracter
j lzo_no           # Repetir el lazo para el siguiente caracter

fin_no:
lw s0, 4(sp)       # Recuperar el valor de los registros de parametros
lw ra, 0(sp)       # Recuperar el valor de la direccion de retorno
addi sp, sp, 8     # Devolver el espacio para cinco registros a la pila
mv a0, a1          # Recupera el valor del puntero
ret               # Terminar la funcion y retornar

mostrar_cadena:
mv a1, a0          # Se mueve el puntero al registro correspondiente
li a0, 4           # Se define el servicio de mostrar cadena
ecall             # Se muestra la cadena por pantalla
ret

```