

**UIT University**

**BACHELOR OF SCIENCE (COMPUTER SCIENCE)**

**CSC-202 COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE**

**PROJECT REPORT**

**TITLE:** Rock Paper Scissor.

**GROUP MEMBERS:**

(Syed Muhammad Abdul Moiz) (23FA-001-CS)

(Muhammad Taha) (23FA-003-CS)

(Prince Irshad) (23FA-035-CS)

**Contents**

[ABSTRACT: 3](#_Toc186585041)

[INTRODUCTION: 3](#_Toc186585042)

[WORKING: 3](#_Toc186585043)

[CODE: 4](#_Toc186585044)

[CONCLUSION: 9](#_Toc186585045)

# **ABSTRACT:**

This document details the implementation of a Rock-Paper-Scissors game using assembly language (MASM). It allows two players to input their choices for each round, evaluates the winner of each round based on the game rules, and announces the overall winner after three rounds. The program demonstrates basic input-output operations, conditional logic, and iterative control flow in assembly language.

# **INTRODUCTION:**

The Rock-Paper-Scissors game is a classic hand game played between two players, where each player simultaneously chooses one of three options: Rock, Paper, or Scissors. The game operates under simple rules:

* Rock beats Scissors.
* Scissors beats Paper.
* Paper beats Rock.

The program simulates this game in assembly language, showcasing the use of registers, memory operations, and interrupts to control program flow and user interaction.

# **WORKING:**

1. **Initialization:**

The program sets up the data and stack segments.

Messages for prompts and results are stored in the .data section.

1. **Input Handling:**

Players input their choices via the keyboard using interrupt 16h.

Input validation ensures the choices are within the range [1, 2, 3].

1. **Game Logic:**

Each player's choice is compared to determine the round winner based on the predefined rules.

Scores are updated accordingly.

1. **Round Iteration:**

The game consists of three rounds. After each round, the remaining rounds counter is decremented.

1. **Final Result:**

At the end of three rounds, the scores are compared to determine the overall winner.

If the scores are tied, the game declares a tie.

1. **Output:**

Results for each round and the final winner are displayed using interrupt 21h.

1. **Program Termination:**

The program gracefully exits using interrupt 21h with function 4Ch.

# **CODE:**

.model small

.stack 100h

.data

titlemsg db "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Welcome! to Rock-Paper-Scissors Game. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$"

msg1 db "Player 1: Enter your choice [1 For Rock] [2 For Paper] [3 For scissors]: $"

msg2 db "Player 2: Enter your choice [1 For Rock] [2 For Paper] [3 For scissors]: $"

tie\_msg db '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Its a tie! \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$'

player1\_wins\_rounds db '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Player 1 wins this round \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$'

player2\_wins\_rounds db '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Player 2 wins this round \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$'

player1\_wins\_whole db '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Player 1 wins! \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$'

player2\_wins\_whole db '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Player 2 wins! \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$'

player1\_choice db ?

player2\_choice db ?

player1\_score db 0

player2\_score db 0

rounds db 3

.code

main proc

mov ax,@data

mov ds,ax

lea dx, titlemsg

mov ah,09h

int 21h

mov dx, 0Ah

mov ah, 02h

int 21h

mov dx, 0Ah

mov ah, 02h

int 21h

mov dx,0DH

mov ah, 02h

int 21h

MOV Cl,rounds

inputs:

lea dx, msg1

mov ah,09h

int 21h

mov ah, 0

int 16h

sub al, '0'

mov player1\_choice,al

mov bl,player1\_choice

mov dx, 0Ah

mov ah, 02h

int 21h

mov dx,0DH

mov ah, 02h

int 21h

cmp bl,1

jl inputs

cmp bl,3

jg inputs

mov dx, 0Ah

mov ah, 02h

int 21h

mov dx,0DH

mov ah, 02h

int 21h

jmp input\_player2

input\_player2:

lea dx, msg2

mov ah,09h

int 21h

mov ah, 0

int 16h

sub al, '0'

mov player2\_choice,al

mov bh,player2\_choice

mov dx, 0Ah

mov ah, 02h

int 21h

mov dx,0DH

mov ah, 02h

int 21h

cmp bh,1

jl input\_player2

cmp bh,3

jg input\_player2

mov dx, 0Ah

mov ah, 02h

int 21h

mov dx,0DH

mov ah, 02h

int 21h

cmp bl,bh

je tiegame

cmp bl,1

je player1\_choice\_rock

cmp bl,2

je player1\_choice\_paper

cmp bl,3

je player1\_choice\_scissors

player1\_choice\_rock:

cmp bh,3

je player1won

cmp bh,2

je player2won

player1\_choice\_paper:

cmp bh,1

je player1won

cmp bh,3

je player2won

player1\_choice\_scissors:

cmp bh,1

je player1won

cmp bh,2

je player2won

player1won:

lea dx, player1\_wins\_rounds

mov ah,09h

int 21h

INC player1\_score

jmp roundover

player2won:

lea dx, player2\_wins\_rounds

mov ah,09h

int 21h

INC player2\_score

jmp roundover

tiegame:

lea dx, tie\_msg

mov ah,09h

int 21h

jmp roundover

roundover:

DEC Cl

mov dx, 0Ah

mov ah, 02h

int 21h

mov dx,0DH

mov ah, 02h

int 21h

mov dx, 0Ah

mov ah, 02h

int 21h

jnz inputs

mov bl,player1\_score

mov bh,player2\_score

cmp bl,bh

je tie

jg player1winner

jl player2winner

tie:

lea dx, tie\_msg

mov ah,09h

int 21h

jmp gameover

player1winner:

lea dx, player1\_wins\_whole

mov ah,09h

int 21h

jmp gameover

player2winner:

lea dx, player2\_wins\_whole

mov ah,09h

int 21h

jmp gameover

gameover:

mov ah, 4Ch

int 21h

main endp

end main

# **CONCLUSION:**

The implementation of Rock-Paper-Scissors in assembly demonstrates foundational skills in programming using low-level language. It covers essential concepts like user input handling, conditional execution, loops, and interrupt-driven I/O. This project serves as an educational example for understanding assembly language and its practical applications.