Homework 4

MIPS Syscalls

Lecturer: Dr. Yisroel Mirsky

Objective:

To design and implement a simple ATM (Automated Teller Machine - כספומט) program in MIPS assembly. This will provide you with hands-on experience with making a program using arithmetic and control flow instructions and syscalls. In this assignment, you should use the MARS MIPS Simulator to write, compile and execute your assembly code.

Background:

ATMs allow bank customers to perform financial transactions, such as depositing money, withdrawing funds, or checking their balance. The current amount of money in the account is called the account `balance’. In this assignment, you'll create a rudimentary ATM system that starts with a fixed starting balance.

Important Note:

- You are provided with an example at the end of the document, use it to check yourself.
    - The user input is indicated with the blue color.
- The Strings printed are very important as the tests are based on them, please ensure you use them correctly (regarding spaces and \n). We advise you to run the example at the end of the file and compare your result with it using a text comparison tool.
    - With that being said, the tests aren’t case-sensitive.

Specifications:

Starting Balance:

Initialize a register with a balance of $1000.

Main Menu:

When the program starts, your ATM program should display the following options to the user:

```
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
```

1. Check Balance:

Display the current balance to the user then return to main menu.

current balance: $1000

2. Deposit Money:

Ask the user to enter the amount they want to deposit. Add this amount to the balance. If the user attempts to deposit more than $5000, display an error message. In all cases, return to the main menu.

enter deposit amount: 100

if the given input is more than 5000, you should print:
“Error: Deposit amount cannot exceed $5000”

If the given input is not a number, print:
“Error: Input is not legal”

3. Withdraw Money:

Ask the user to enter the amount they wish to withdraw. Subtract this amount from the balance. If the user tries to withdraw more than their current balance, display an error message. Also, set a withdrawal limit of $500 per transaction. Finally, return to main menu.

enter withdrawal amount: 100

if withdrawal isn’t legal, print:
“Error: Insufficient funds or withdrawal limit exceeded”

If the given input is not a number, print:
“Error: Input is not legal”

Exit:

Display a thank you message and terminate the program.
"Thank you for using the ATM. Goodbye!"

Additional Requirements:

- Use appropriate labels to make your code clear and readable.

  Handle basic input errors, such as entering non-numeric values for deposit or withdrawal amounts. If the given input is not a number, print: “Error: Input is not legal”
- Comment your code to explain critical sections. Commented lines in assembly start with the ‘#’ symbol

Bonus Task (Optional) +5 points:

Implement a PIN (Personal Identification Number) which will be checked at the start of your ATM program. Allow the user three attempts to enter the correct PIN. If they fail three times, display an error message and exit the program. Assume a fixed PIN like 1234 for this exercise.

Tips:

- Remember to repeatedly display the main menu after each operation until the user chooses to exit.
- Utilize loops for repeated tasks or checks.
- Use branch instructions to handle different user choices effectively.

Submission:

Submit your MIPS assembly code with the filename main.asm. Your code will be tested for cheating (copied code) using anti-plagiarism software.

Evaluation Criteria:

Your code will be evaluated based on the following:

- Correctness of the program.
- Clarity and organization of the code.
- Proper use of arithmetic and control flow instructions.
- Handling edge cases and input errors.

Some Examples(Bonus Examples below):

```
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
1
Current Balance: $1000
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
2
Enter deposit amount: 100
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
1
Current Balance: $1100
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
2
Enter deposit amount: 600
Main Menu:
1. Check Balance
2. Deposit Money
```

```
3. Withdraw Money
4. Exit
1
Current Balance: $1700
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
2
Enter deposit amount: 5500
Error: Deposit amount cannot exceed $5000
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
3
Enter withdrawal amount: 600
Error: Insufficient funds or withdrawal limit exceeded
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
3
Enter withdrawal amount: 2000
Error: Insufficient funds or withdrawal limit exceeded
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
4
```

```
Thank you for using the ATM. Goodbye!

-- program is finished running --
```

Bonus Examples (After The pin part, they are exactly like the previous one):

```
Enter PIN: 1234
Main Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
4
Thank you for using the ATM. Goodbye!

-- program is finished running --
```

```
Enter PIN: 4567
Error: Incorrect PIN
Enter PIN: 4555
Error: Incorrect PIN
Enter PIN: 4555
Error: Incorrect PIN
Error: No attempts remaining. Exiting program

-- program is finished running --
```

```
Enter PIN: 4566
Error: Incorrect PIN
Enter PIN: 4566
Error: Incorrect PIN
Enter PIN: 1234
```

```
        Main Menu:
        1. Check Balance
        2. Deposit Money
        3. Withdraw Money
        4. Exit
        4
        Thank you for using the ATM. Goodbye!

-- program is finished running --
```