Experiment 12: Building a Rule-Based Expert System using Shell Scripting

Name = Priyadarshi Prabhakar SAP ID 590029237

Theory

- **Process Automation and Job Scheduling:** Automating repetitive system tasks using shell scripts.
- **System Administration Scripts:** Managing essential services, daemons, and system operations.
- **Rule-Based Expert System:** A simple AI-like system that makes decisions based on predefined rules.

Objective

To develop a rule-based expert system using shell scripting that provides recommendations based on user symptoms.

Instructions

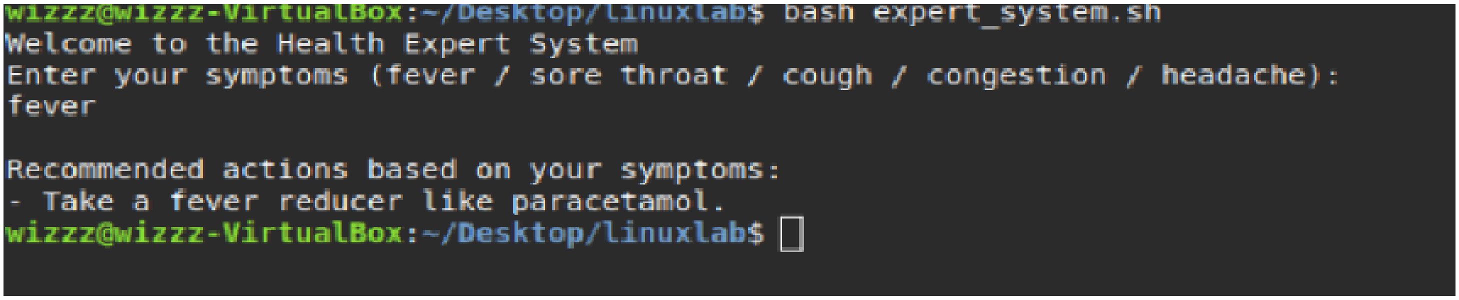
1. Create a shell script named `expert_system.sh`.
2. Implement rules using `if-elif-else` conditions.
3. Example rules:
    - If the user has **fever**, recommend taking fever reducer medication.
    - If they have a **sore throat**, recommend gargling with warm salt water.
    - If they have **cough + congestion**, recommend warm fluids and cough syrup.
4. Prompt the user to enter their symptoms.
5. Evaluate rules and display recommendations.
6. If no rules match, display a general suggestion.
7. Test the script with multiple symptom inputs.

8. Modify or expand rules as needed.
9. Document logic and rules inside the script.
10. Write a summary about challenges, observations, and improvements.

Sample Shell Script (`expert_system.sh`)

```bash
#!/bin/bash

echo "Welcome to the Health Expert System"
echo "Enter your symptoms (fever / sore throat / cough / congestion / headache): "
read symptoms

recommendation=""

if [[ $symptoms == *"fever"* ]]; then
    recommendation="$recommendation
- Take a fever reducer like paracetamol."
fi

if [[ $symptoms == *"sore throat"* ]]; then
    recommendation="$recommendation
- Gargle with warm salt water."
fi

if [[ $symptoms == *"cough"* && $symptoms == *"congestion"* ]]; then
    recommendation="$recommendation
- Drink warm fluids and take cough syrup."
fi

if [[ $symptoms == *"headache"* ]]; then
    recommendation="$recommendation
- Rest in a quiet room and stay hydrated."
fi

if [[ -z "$recommendation" ]]; then
    echo -e "
No specific match found. Please consult a doctor if symptoms persist."
else
    echo -e "
Recommended actions based on your symptoms:$recommendation"
fi
```

OUTPUT



```
wizzz@wizzz-VirtualBox:~/Desktop/linuxlab$ bash expert_system.sh
Welcome to the Health Expert System
Enter your symptoms (fever / sore throat / cough / congestion / headache):
fever

Recommended actions based on your symptoms:
- Take a fever reducer like paracetamol.
wizzz@wizzz-VirtualBox:~/Desktop/linuxlab$ 
```

Documentation of Logic & Rules

Symptom	Rule	Recommendation
Fever	`if [[ $symptoms == *"fever"* ]]`	Take fever reducer
Sore throat	Matches phrase "sore throat"	Gargle with warm salt water
Cough + congestion	Checks for both terms	Warm fluids + cough syrup
Headache	Matches "headache"	Rest + hydration
No match	Default case	General suggestion

Summary Report

Challenges Faced

- Designing clear rules that avoid overlapping conditions.
- Handling multiple symptoms entered together.
- Ensuring user input matches rule patterns accurately.

Observations

- Rule-based systems behave deterministically.
- Adding more rules increases accuracy but also complexity.
- Simple text-matching can still build a useful expert system.

Improvements

- Add menus instead of free text input.
- Use case statements for cleaner logic.
- Implement multi-rule scoring to give prioritized recommendations.
- Expand expert system to other fields (e.g., system diagnostics, agriculture advice, etc.).

End of Document