

UNISHELL

Project Report

Group members: – Hamza Zarah & Albert Jasari
Date: – 10.06.2023
Course Name: – Operating System, Spring 2023

Please enter a command:

1. The Introduction

1.1 Motivation

This report is about our UniShell project. We, Hamza Zarah and Albert Jasari started this project together with the aim of addressing a issue observed in users of all age groupe – the excessive and sometime unproductive use of personal computers and laptops.

In the digital world as we know it today, our lives are inextricably linked with technology. Above all, computers are fundamental tools in various areas of life, such as at work or, in our case, in studies. However, the omnipresence of computers and our dependence on them have led to some challenges that can negatively impact our health and productivity. Sitting for extended periods in front of the computer, insufficient hydration due to distraction from computer work, and inefficient time management are common issues that many people, including students, struggle with.

Considering these problems, the following questions arise:

1. How can we remind computer users to take the necessary breaks to reduce the risk of eye strain from digital devices and sedentary activities?
2. How can we motivate students to use their time more effectively?
3. How can we promote regular water intake to prevent dehydration during prolonged computer use?

2.2 Objective

The goal of our project is to implement UniShell, a personal productivity assistant Shell that provides the user with timely reminders to take breaks, hydrate and move around. These reminders are not just randomly timed, they are based on proven techniques like the Pomodoro technique that helps maintain productivity while also preventing burnout.

2. Background

Several concepts, techniques, and technologies are essential in addressing the problem of excessive use of computers and the associated health and productivity challenges. These include the Pomodoro Technique, hydration reminders, and movement reminders, all of which are part of UniShell's core functionality.

2.1 Pomodoro Technique

The Pomodoro Technique is a time-management technique invented by Francesco Cirillo in the late 1980s. The strategy uses a timer to divide work into intervals of 25 minutes, separated by short breaks. The strategy seeks to boost productivity and focus by ensuring regular brain rest, reducing cognitive overload, and allowing good task management.

2.2 Hydration Reminders

Hydration reminders are an important tool for preventing dehydration. Dehydration can lead to a decrease in physical and cognitive performance, with symptoms including fatigue, dizziness, and confusion. Given that computer work can often be absorbing, causing individuals to forget about their water intake, the role of hydration reminders is crucial in maintaining the optimal cognitive function needed for productivity.

2.3 Movement reminders

Reminders to move are meant to reduce the negative effects of sitting for too long and other sedentary behaviors, which are known to lead to tension and back pain. Sitting for extended periods of time can cause stiffness and strain on your muscles and joints, especially in your neck and back. Regular physical activity can help ease these discomforts and improve overall musculoskeletal health, even if it's just standing up and moving around for a while.

4. Implementation, Configuration and Setup

4.1 Implementation

The implementation of UniShell can be broken down into three primary segments - the Pomodoro Timer, Hydration Reminder, and the Movement Reminder. All three modules make use of C's multithreading capability to run concurrently and independently.

The Pomodoro Timer implements the Pomodoro Technique. The timer ticks down from a predefined 25 minute interval, and at the end of the interval, it triggers a little notification to the user indicating it's time to take a short break. This timer employs C's pthread library for creating a separate thread that runs concurrently with the main program. In the event of a new timer request, signal handling is used to interrupt the ongoing timer and start a fresh one. [A code snippet of the thread creation is as follows:](#)

```
if (pthread_create(&pomodoro_thread, NULL, pomodoro_thread_func, NULL) != 0)
{
    fprintf(stderr, "Error creating pomodoro thread\n");
}
```

The Hydration Reminder and Movement Reminder follow a similar pattern, running on independent threads and triggering alarms at predefined intervals to remind the user to drink water or move around, respectively.

The user interacts with UniShell through a simple, text-based command-line interface. The interface supports a variety of commands for starting and stopping timers. At the start of the application, the user is presented with a list of available commands, providing a self-explanatory and intuitive user experience.

For error handling and exception management, comprehensive checks were implemented at all potential points of failure. [A simple example is, when processing user input, we ensure that the input is not null before proceeding:](#)

```
if (args[0] == NULL)
{
    printf("Please enter a command\n");
}
```

This implementation ensures that the program remains robust and responsive even in the event of unexpected inputs or errors, enhancing the overall user experience and reliability of UniShell.

4.2 Configuration and Setup

To use UniShell, follow the steps below:

1. Clone the repository from GitHub.
2. Navigate to the project directory: `cd UniShell/src`
3. Run make command to compile the project: `make`
4. Now, you can run the UniShell executable: `make run`

When you start the application, you will be presented with a list of available commands, which is a self-explanatory and intuitive.

5. Results

The results of the UniShell project are observed directly in the terminal interface when the user interacts with the shell. After successfully launching UniShell, the user is first greeted

with a friendly welcome message along with a list of available commands. This helps to create an intuitive and user-friendly environment right from the start.



```

UNISHELL

Welcome to Your Personal Assistant

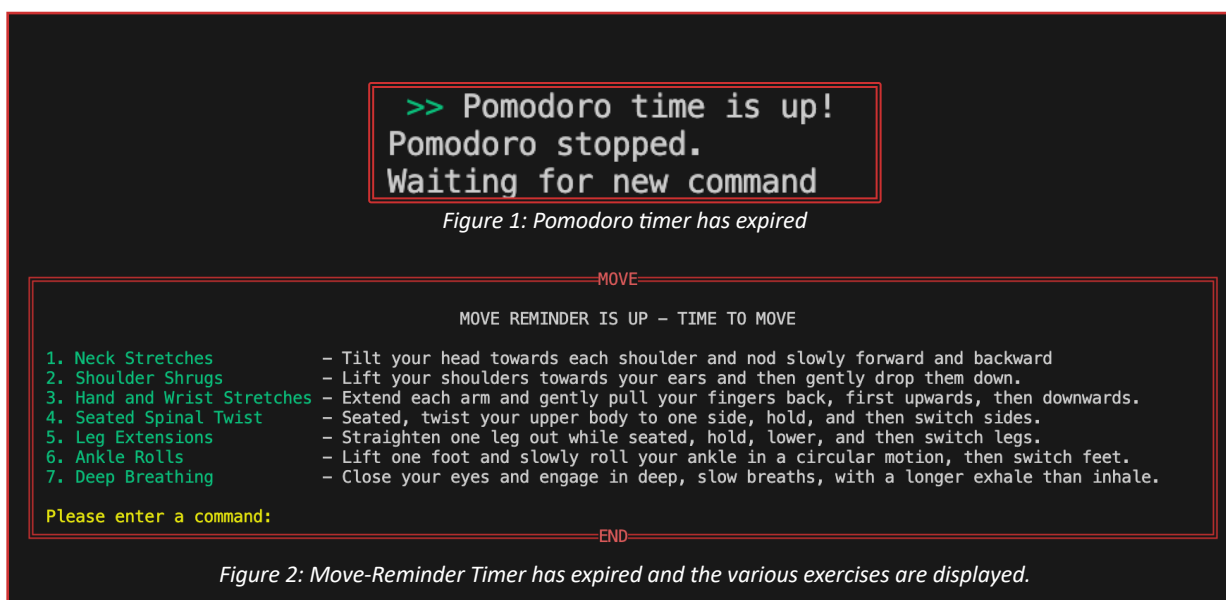
This program assists you in maintaining regular breaks
and reminds you to drink enough water and move around.

Commands:
start_pomodoro - Start the Pomodoro Timer
stop_pomodoro  - Stop the Pomodoro Timer
start_water    - Start the Water Reminder Timer
stop_water     - Stop the Water Reminder Timer
start_move     - Start the Movement Reminder Timer
stop_move      - Stop the Movement Reminder Timer
start_all      - Start all timers
stop_all       - Stop all timers
exit           - Exit the program
help           - Show this help again

Please enter a command:
```

When the user starts working and starts the timers, UniShell works unobtrusively in the background and starts various or all reminder threads based on the predefined intervals, depending on the preference. After each timer expires, UniShell warns the user with an appropriate reminder within the terminal interface.

For example, when the set interval for the Pomodoro technique timer expires, the shell displays a message prompting the user to take a break from work. A similar prompt is displayed when the reminder times for hydration or exercise are reached, prompting the user to drink water or do a short physical exercise, respectively.



```

>> Pomodoro time is up!
Pomodoro stopped.
Waiting for new command

Figure 1: Pomodoro timer has expired

MOVE
MOVE REMINDER IS UP - TIME TO MOVE

1. Neck Stretches      - Tilt your head towards each shoulder and nod slowly forward and backward
2. Shoulder Shrugs     - Lift your shoulders towards your ears and then gently drop them down.
3. Hand and Wrist Stretches - Extend each arm and gently pull your fingers back, first upwards, then downwards.
4. Seated Spinal Twist - Seated, twist your upper body to one side, hold, and then switch sides.
5. Leg Extensions      - Straighten one leg out while seated, hold, lower, and then switch legs.
6. Ankle Rolls          - Lift one foot and slowly roll your ankle in a circular motion, then switch feet.
7. Deep Breathing       - Close your eyes and engage in deep, slow breaths, with a longer exhale than inhale.

Please enter a command:

END

Figure 2: Move-Reminder Timer has expired and the various exercises are displayed.
```

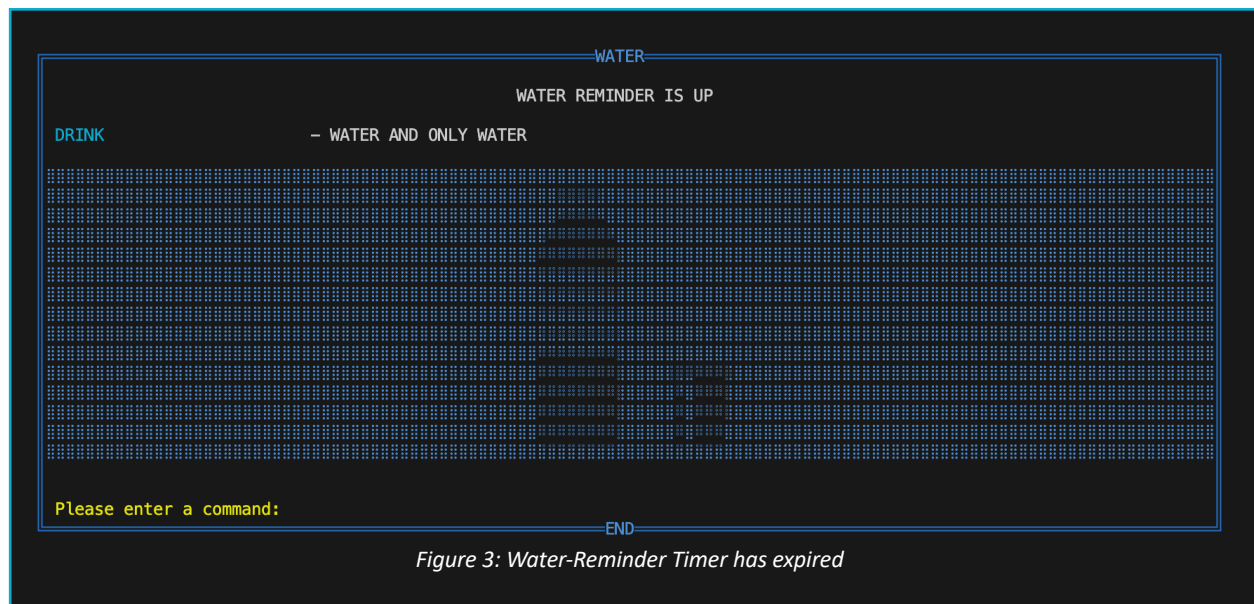


Figure 3: Water-Reminder Timer has expired

6. Discussion

6.1 Problems

In the early stages of the project, building the foundation on which the rest of the development would be built was a major challenge. We were initially unsure where to start. Creating the shell was a complex process, and at the beginning of the project our experience with C programming was limited. Nevertheless, we are pleased to present a final result that demonstrates our efforts.

The development of the shell was not without difficulties. Implementing multithreading to process the different memories simultaneously and ensure that they worked seamlessly together required a deep understanding of the C programming language and its threading library.

In addition, signal processing to properly manage user input and system signals was another complex task. Ensuring that the system worked smoothly without crashing and disrupting the user's workflow was a challenge, but one that ultimately paid off.

6.2 Improvements

Despite the challenges encountered, we have conducted a critical review of our work and identified areas for improvement in potential future iterations of the UniShell project.

1. Timer customisation: Currently, the duration of timers in UniShell is hard-coded. This lack of flexibility could prove inconvenient for users with different productivity schedules. To make UniShell more customisable, a future enhancement could include a feature that allows users to adjust the duration of timers based on their personal needs.

2. External notifications: Currently, all notifications from UniShell are displayed within the terminal. However, the user may not always be paying attention to the terminal and therefore miss important reminders. A future improvement could include notifications appearing outside the terminal to ensure users are informed when a timer has expired, regardless of their current focus on the computer.
3. Personalised reminders: UniShell currently works with a fixed set of reminders. To improve the usability and versatility of UniShell, future versions of UniShell could allow users to customise these reminders as well as their intervals. This would allow users to customise the functionality of the tool to their individual productivity needs and habits.

7. Conclusion & Lessons Learned

Despite the initial challenges, we have successfully created a functional program that not only meets our initial expectations, but also provides a solid foundation for future development. Through the creation of the UniShell and, of course, the Operating Systems lecture, we have gained a deeper understanding of C programming, multithreading and signal processing. This knowledge and experience, along with the feedback and insights we have gained, will be able to allow us (if we have time) to refine and improve UniShell in the future, with the goal of long-term success.

Although this project has been challenging, it has been an invaluable learning experience. Not only did it improve our technical skills, but it also taught us the importance of perseverance and critical evaluation in software development.