GEBZE
TEKNİK ÜNİVERSİTESİ

CSE344 - Homework#2

Muhammet Talha Memişoğlu

210104004009

Introduction

The object of homework is implementing a communication protocol between two processes using IPC and creating a daemon process to handle background operations.. This program was developed in C and tested in Debian11.

Code Explanation

Program takes two integer from the command line. Exits with usage info if not provided.

FIFO Creation

Creates three FIFOs:

- FIFO1: Parent → Child 1 (for sending two integers)
- FIFO2: Child 1 → Child 2 (sends the larger number)
- DAEMON_FILE: Parent/Signal handler → Daemon (logs processes’ termination status)

Parent process creates two child using fork().

Child 1

- Sleeps for 10 seconds.
- Reads two integers from FIFO1.
- Determines the larger number.
- Sends the result to FIFO2.
- Exit

Child 2

- Sleeps for 10 seconds.
- Reads the larger number from FIFO2.
- Prints it to terminal.
- Exit

Parent initialize the daemon via daemonize().

While parent process is waiting for childs, it prints proceeding message.

Writes the two integers to FIFO1.

Sets up the SIGCHLD handler with handle_sigchld().

- In the handle_sigchld() function, the parent process waits for the termination of its child processes. Once a child exits, the parent sends its termination status to the daemon for logging via a separate FIFO. Additionally, the use of waitpid() ensures that **zombie processes** are properly reaped.

Waits until both children have terminated (tracked via child_counter).

Daemon Process

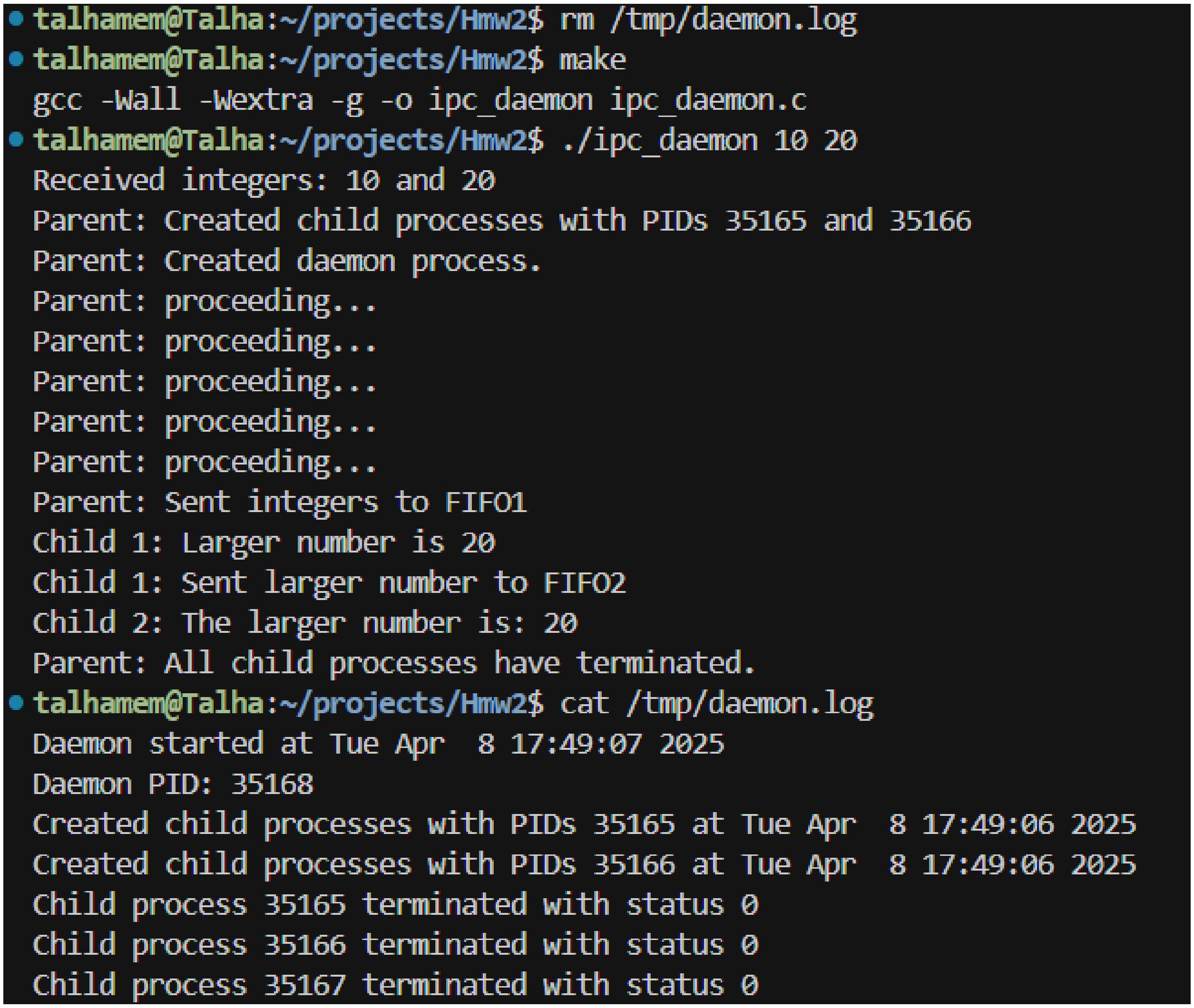
- Daemon starts with usual steps.
    - Fork and let the parent exit.
    - Child becomes session leader.
    - Fork again to ensure the daemon can’t reacquire a terminal.
    - Let the parent exit.
    - Change working directory to root.
    - Close all open file descriptors.
- Redirect standard file descriptors to log file using dup2() function.
- Handles SIGUSR1, SIGHUP, SIGTERM, SIGCHLD signals with handle_daemon_signal.
- Log daemon startup.
- Logs PIDs and creation times of child processes.
- Opens DAEMON_FILE (FIFO which is used by handle_sigchld()) and reads messages sent by handle_sigchld().
- Logs **exit statuses** of all processes. (There is extra process in this log because of forking again to ensure daemon can’t reacquire a terminal in daemon process)
- Run monitor_children() function to handle timeouts for child processes.
- Continuously logs these messages to stdout (redirected to daemon.log).

Interprocess Communication Flow

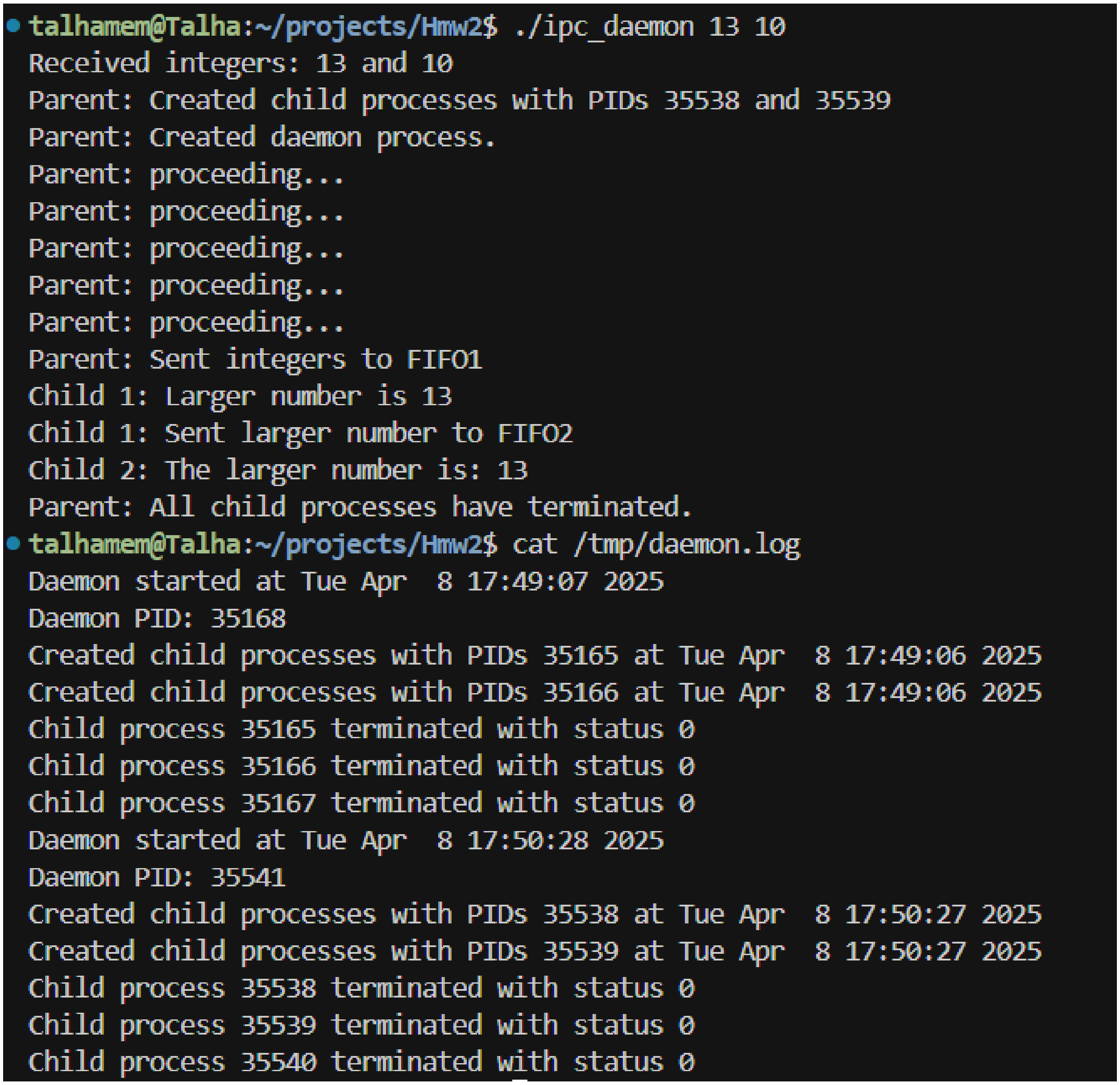
1. **Parent → Child 1** via FIFO1: Sends two integers.

2. **Child 1** → **Child 2** via FIFO2: Sends the larger number.
3. **Children** → **Daemon** via DAEMON_FILE: Termination messages.
4. **Daemon** logs everything to /tmp/daemon.log.

Screenshots



```
talhamem@Talha:~/projects/Hmw2$ rm /tmp/daemon.log
talhamem@Talha:~/projects/Hmw2$ make
gcc -Wall -Wextra -g -o ipc_daemon ipc_daemon.c
talhamem@Talha:~/projects/Hmw2$ ./ipc_daemon 10 20
Received integers: 10 and 20
Parent: Created child processes with PIDs 35165 and 35166
Parent: Created daemon process.
Parent: proceeding...
Parent: proceeding...
Parent: proceeding...
Parent: proceeding...
Parent: proceeding...
Parent: Sent integers to FIFO1
Child 1: Larger number is 20
Child 1: Sent larger number to FIFO2
Child 2: The larger number is: 20
Parent: All child processes have terminated.
talhamem@Talha:~/projects/Hmw2$ cat /tmp/daemon.log
Daemon started at Tue Apr  8 17:49:07 2025
Daemon PID: 35168
Created child processes with PIDs 35165 at Tue Apr  8 17:49:06 2025
Created child processes with PIDs 35166 at Tue Apr  8 17:49:06 2025
Child process 35165 terminated with status 0
Child process 35166 terminated with status 0
Child process 35167 terminated with status 0
```



```
• talhamem@Talha:~/projects/Hmw2$ ./ipc_daemon 13 10
  Received integers: 13 and 10
  Parent: Created child processes with PIDs 35538 and 35539
  Parent: Created daemon process.
  Parent: proceeding...
  Parent: proceeding...
  Parent: proceeding...
  Parent: proceeding...
  Parent: proceeding...
  Parent: Sent integers to FIFO1
  Child 1: Larger number is 13
  Child 1: Sent larger number to FIFO2
  Child 2: The larger number is: 13
  Parent: All child processes have terminated.
• talhamem@Talha:~/projects/Hmw2$ cat /tmp/daemon.log
  Daemon started at Tue Apr  8 17:49:07 2025
  Daemon PID: 35168
  Created child processes with PIDs 35165 at Tue Apr  8 17:49:06 2025
  Created child processes with PIDs 35166 at Tue Apr  8 17:49:06 2025
  Child process 35165 terminated with status 0
  Child process 35166 terminated with status 0
  Child process 35167 terminated with status 0
  Daemon started at Tue Apr  8 17:50:28 2025
  Daemon PID: 35541
  Created child processes with PIDs 35538 at Tue Apr  8 17:50:27 2025
  Created child processes with PIDs 35539 at Tue Apr  8 17:50:27 2025
  Child process 35538 terminated with status 0
  Child process 35539 terminated with status 0
  Child process 35540 terminated with status 0
```

Conclusion

In this project, I had a hard time finding a way for intercommunication between handle_sigchld() and daemon to write exit statuses of processes. I was helped by AI tools and websites in that point. Additionally, debugging was hard so I put some functions and track from terminal to handle these bugs. Finally, I have managed to complete this homework, and all requirements and bonuses are fulfilled properly.