PA3 Report – Seat Reservation System

1. Project Goal

The purpose of this project was to create a multi-user seat reservation system that enables clients to book, query, and cancel seats through a client-server model. The server supports concurrent client connections using multi-threading, and the client provides both interactive and file-driven ways for users to interact with the reservation system, as required by the assignment.

---

2. Design & Implementation

On the server side, I maintained 100 seats and managed up to 10,000 users, using threads to handle multiple clients at the same time. Each seat had its own lock to guarantee data consistency when accessed by several users. Users are automatically registered upon their first login, and passwords are securely managed with Argon2 hashing. Duplicate logins from separate clients are blocked to maintain session integrity. All server-client communications follow a custom protocol and type-length-value encoding, which ensures flexible and safe data transfer.

The client program connects to the server via IP and port, supporting both command-line input and file-based automation. All required actions (login, booking, querying, canceling, confirming bookings, logout, and termination) are implemented, and the client displays clear results and error messages as defined in the assignment spec.

---

3. Example Usage & Testing

To verify the functionality, I ran through all the main user actions such as logging in, booking seats, checking seat information, and logging out. I also tested various error conditions, like attempting to book without logging in, trying to reserve out-of-range seats, or making duplicate login attempts. In each case, the client provided appropriate feedback to the user according to the assignment's required message formats.

I included **screenshots (ss)** at the end of this report that show these interactions in action. The screenshots illustrate the server running, the client performing both normal and error-case

commands, and the way feedback is displayed to users. These images serve as visual evidence that all required behaviors and edge cases are handled correctly.

---

4. Resource Management

All dynamically allocated resources, such as user records, seat data, threads, and locks, are properly freed or destroyed on program exit. I checked this using Valgrind, which confirmed there were no memory leaks, as required by the assignment.

---

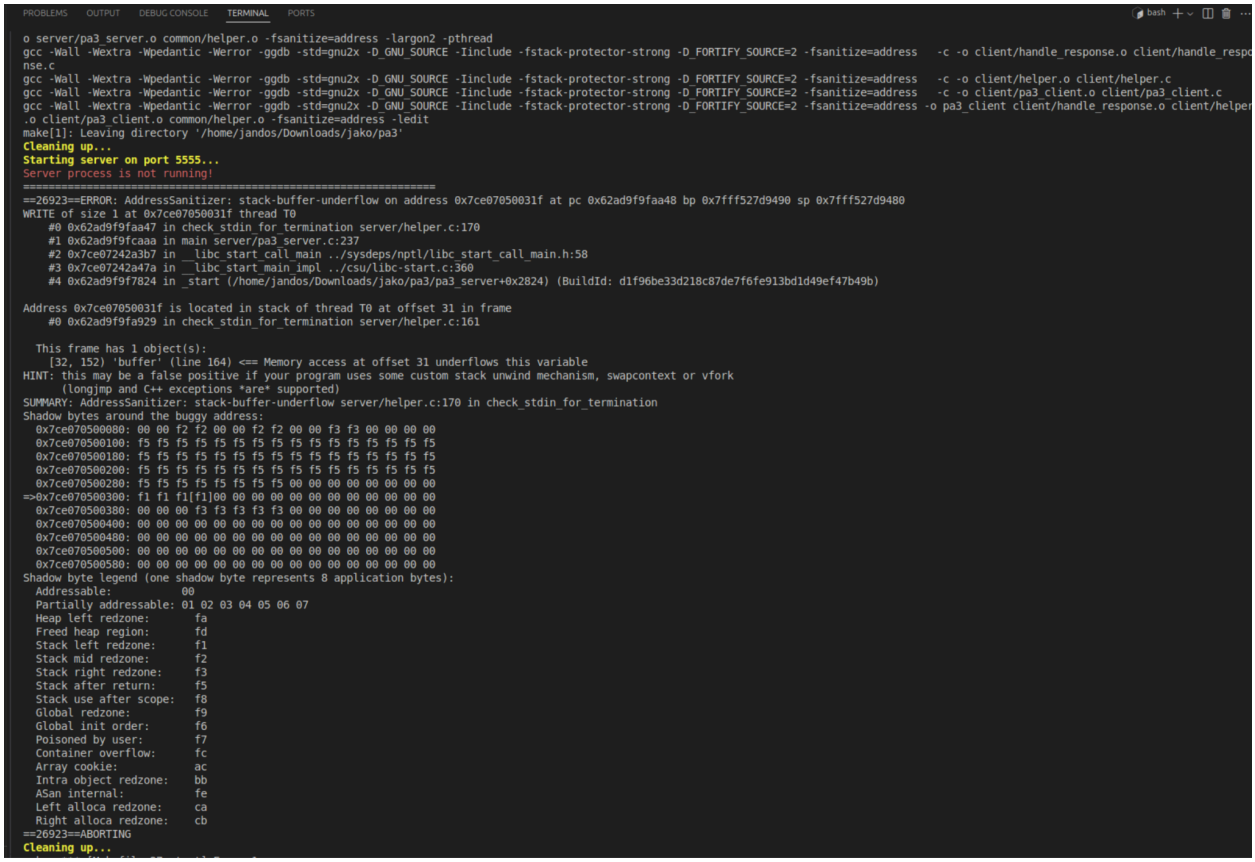
5. How to Run

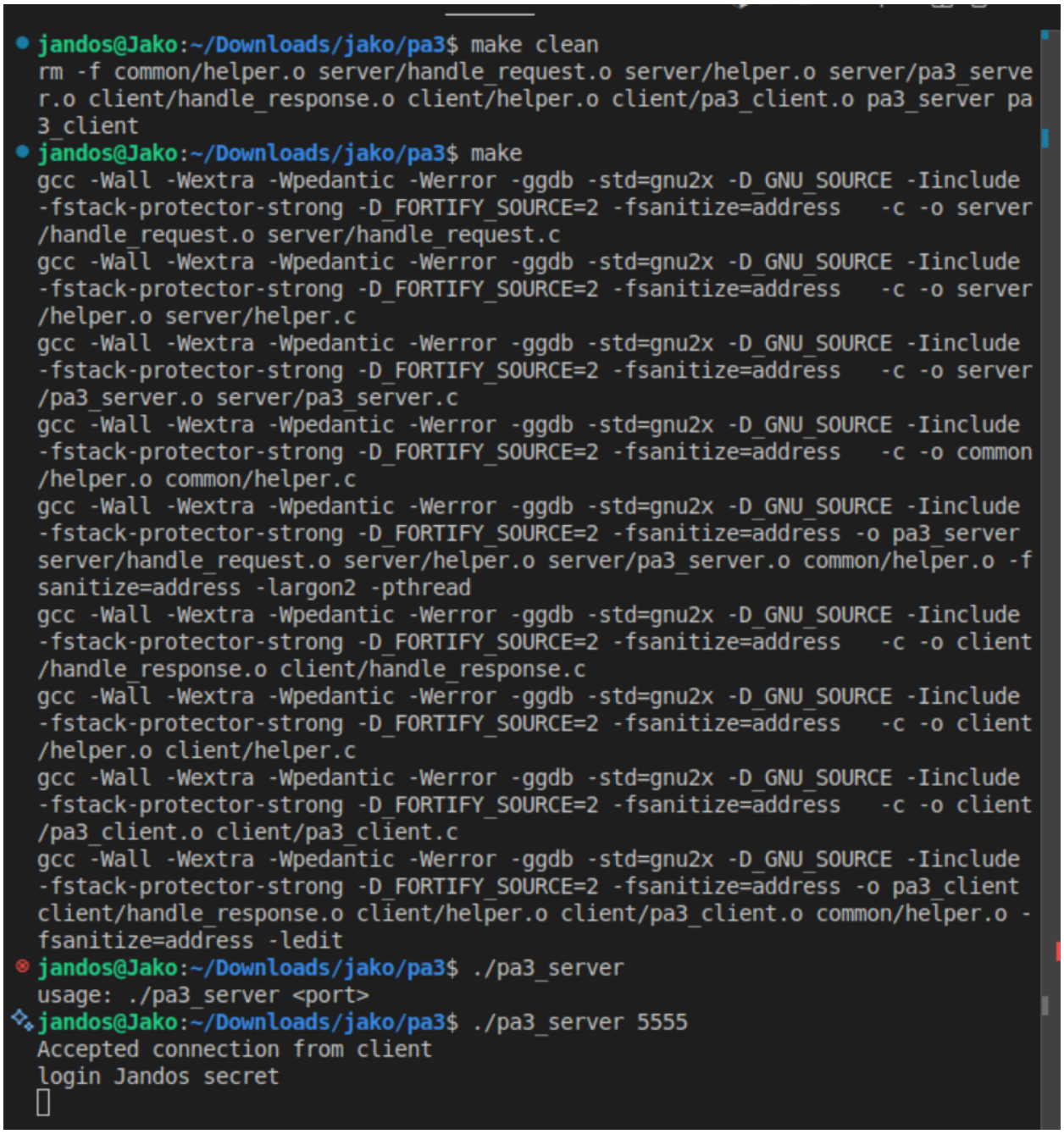
The system can be built and run using `make`, followed by launching the server with a chosen port, and connecting with the client using the appropriate address and port. Both interactive command entry and file-based input are supported.

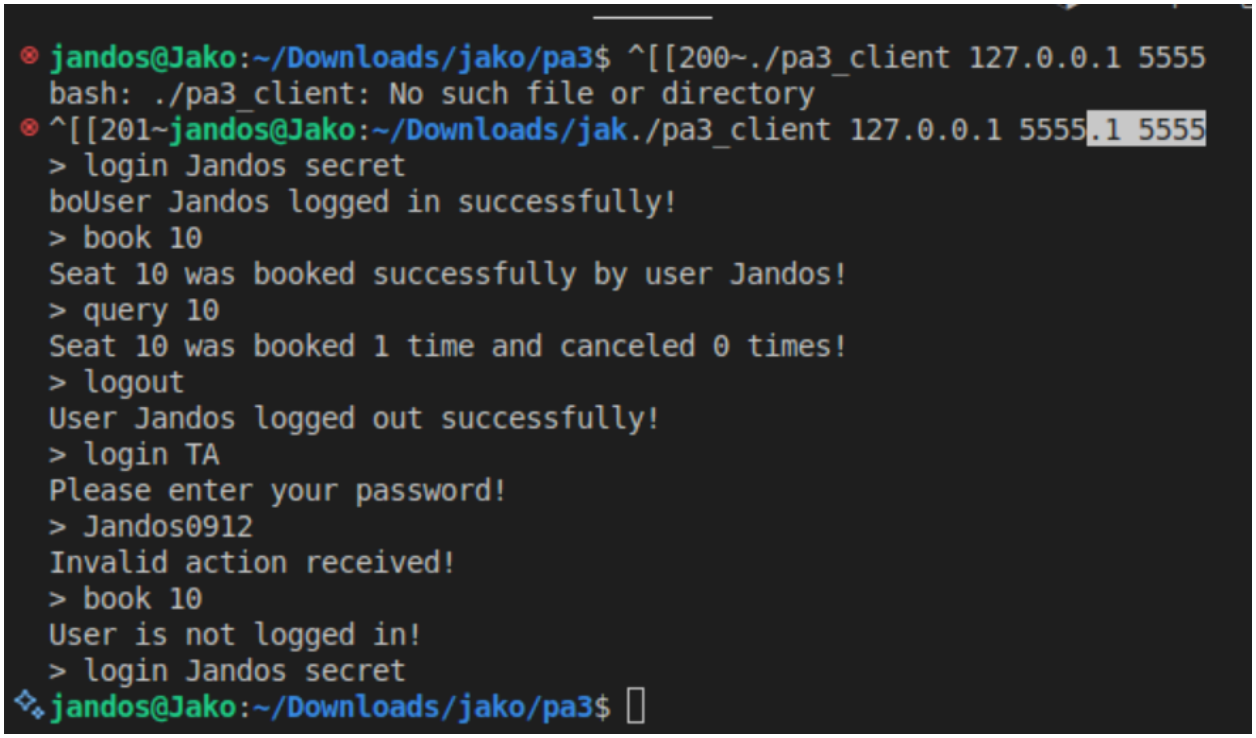
---

6. Conclusion

This project provided practical experience with network programming, concurrency, and data synchronization. My solution meets all assignment requirements for functionality, robustness, and usability, as demonstrated by the included screenshots.


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
bash
o server/pa3_server.o common/helper.o -fsanitize=address -largon2 -pthread
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o client/handle_response.o client/handle_respo
nse.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o client/helper.o client/helper.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o client/pa3_client.o client/pa3_client.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -o pa3_client client/handle_response.o client/helper
.o client/pa3_client.o common/helper.o -fsanitize=address -ledit
make[1]: Leaving directory '/home/jandos/Downloads/jako/pa3'
Cleaning up...
Starting server on port 5555...
Server process is not running!
==26923==ERROR: AddressSanitizer: stack-buffer-underflow on address 0x7ce07050031f at pc 0x62ad9f9faa48 bp 0x7fff527d9490 sp 0x7fff527d9480
WRITE of size 1 at 0x7ce07050031f thread T0
#0 0x62ad9f9faa47 in check_stdin_for_termination server/helper.c:170
#1 0x62ad9f9fcaaa in main server/pa3_server.c:237
#2 0x7ce07242a3b7 in __libc_start_call_main ../sysdeps/nptl/libc_start_call_main.h:58
#3 0x7ce07242a47a in __libc_start_main_impl ../csu/libc-start.c:360
#4 0x62ad9f9f7824 in _start (/home/jandos/Downloads/jako/pa3/pa3_server+0x2824) (BuildId: d1f96be33d218c87de7f6fe913bd1d49ef47b49b)
Address 0x7ce07050031f is located in stack of thread T0 at offset 31 in frame
#0 0x62ad9f9fa929 in check_stdin_for_termination server/helper.c:161
This frame has 1 object(s):
[32, 152) 'buffer' (line 164) <== Memory access at offset 31 underflows this variable
HINT: this may be a false positive if your program uses some custom stack unwind mechanism, swapcontext or vfork
(longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-underflow server/helper.c:170 in check_stdin_for_termination
Shadow bytes around the buggy address:
0x7ce070500080: 00 00 f2 f2 00 00 f2 f2 00 00 f3 f3 00 00 00 00
0x7ce070500100: f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5
0x7ce070500180: f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5
0x7ce070500200: f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5 f5
0x7ce070500280: f5 f5 f5 f5 f5 f5 f5 f5 00 00 00 00 00 00 00 00
=>0x7ce070500300: f1 f1 f1[f1]00 00 00 00 00 00 00 00 00 00 00 00
0x7ce070500380: 00 00 00 f3 f3 f3 f3 f3 00 00 00 00 00 00 00 00
0x7ce070500400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x7ce070500480: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x7ce070500500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x7ce070500580: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone: fa
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Array cookie: ac
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
==26923==ABORTING
Cleaning up...


jandos@Jako:~/Downloads/jako/pa3$ make clean
rm -f common/helper.o server/handle_request.o server/helper.o server/pa3_server.o client/handle_response.o client/helper.o client/pa3_client.o pa3_server pa3_client
jandos@Jako:~/Downloads/jako/pa3$ make
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o server/handle_request.o server/handle_request.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o server/helper.o server/helper.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o server/pa3_server.o server/pa3_server.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o common/helper.o common/helper.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -o pa3_server server/handle_request.o server/helper.o server/pa3_server.o common/helper.o -fsanitize=address -largon2 -pthread
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o client/handle_response.o client/handle_response.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o client/helper.o client/helper.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -c -o client/pa3_client.o client/pa3_client.c
gcc -Wall -Wextra -Wpedantic -Werror -ggdb -std=gnu2x -D_GNU_SOURCE -Iinclude -fstack-protector-strong -D_FORTIFY_SOURCE=2 -fsanitize=address -o pa3_client client/handle_response.o client/helper.o client/pa3_client.o common/helper.o -fsanitize=address -ledit
jandos@Jako:~/Downloads/jako/pa3$ ./pa3_server
usage: ./pa3_server <port>
jandos@Jako:~/Downloads/jako/pa3$ ./pa3_server 5555
Accepted connection from client
login Jandos secret


jandos@Jako:~/Downloads/jako/pa3$ ^[[200~./pa3_client 127.0.0.1 5555
bash: ./pa3_client: No such file or directory
^[[201~jandos@Jako:~/Downloads/jak./pa3_client 127.0.0.1 5555.1 5555
> login Jandos secret
boUser Jandos logged in successfully!
> book 10
Seat 10 was booked successfully by user Jandos!
> query 10
Seat 10 was booked 1 time and canceled 0 times!
> logout
User Jandos logged out successfully!
> login TA
Please enter your password!
> Jandos0912
Invalid action received!
> book 10
User is not logged in!
> login Jandos secret
jandos@Jako:~/Downloads/jako/pa3$