

گزارش کار شماره ۴ آزمایش سیستم عامل

نام و نام خانوادگی:

محمد مهدی حائری 40131008

محمد علی آقایی 40131059

گزارش تمرین سیستم عامل: ارتباط بین فرایندها با حافظه اشتراکی

هدف: ایجاد دو فرآیند (تولید کننده و مصرف کننده) که از طریق حافظه اشتراکی با یکدیگر ارتباط برقرار کنند.

شرح:

- فرآیند تولید کننده:

- اعداد تصادفی تولید کرده و در حافظه اشتراکی ذخیره می‌کند.

- فرآیند مصرف کننده:

- اعداد موجود در حافظه اشتراکی را خوانده و مجموع آن‌ها را محاسبه می‌کند.

ملاحظات:

- برای ایجاد حافظه اشتراکی در macOS، از توابع `shm_open`، `mmap` (POSIX) و ... استفاده می‌شود.

- به منظور جلوگیری از تداخل در دسترسی به حافظه اشتراکی، از مکانیزم‌های همگام‌سازی (مانند سمافورها) استفاده می‌شود.

- برای تولید اعداد تصادفی از تابع `rand` استفاده می‌شود.

گزارش تمرین سیستم عامل: سیستم مدیریت انبار پیشرفته

هدف: پیاده‌سازی یک سیستم مدیریت انبار پیشرفته با قابلیت پشتیبانی از چندین کاربر همزمان و انتقال اجناس بین کاربران.

ویژگی‌های کلیدی:

- پشتیبانی از چندین کاربر همزمان با استفاده از نخ‌ها (threads).
- نگهداری لیست اجناس جداگانه برای هر کاربر.
- امکان انتقال اجناس بین کاربران با استفاده از دستور `send [target_client_name] [[product_name] [amount`
- بررسی موجودی کافی قبل از انتقال اجناس.
- اضافه کردن جنس به لیست کاربر مقصد در صورت عدم وجود.
- جلوگیری از انتقال با مقدار منفی.
- جلوگیری از ایجاد کاربران با نام تکراری.

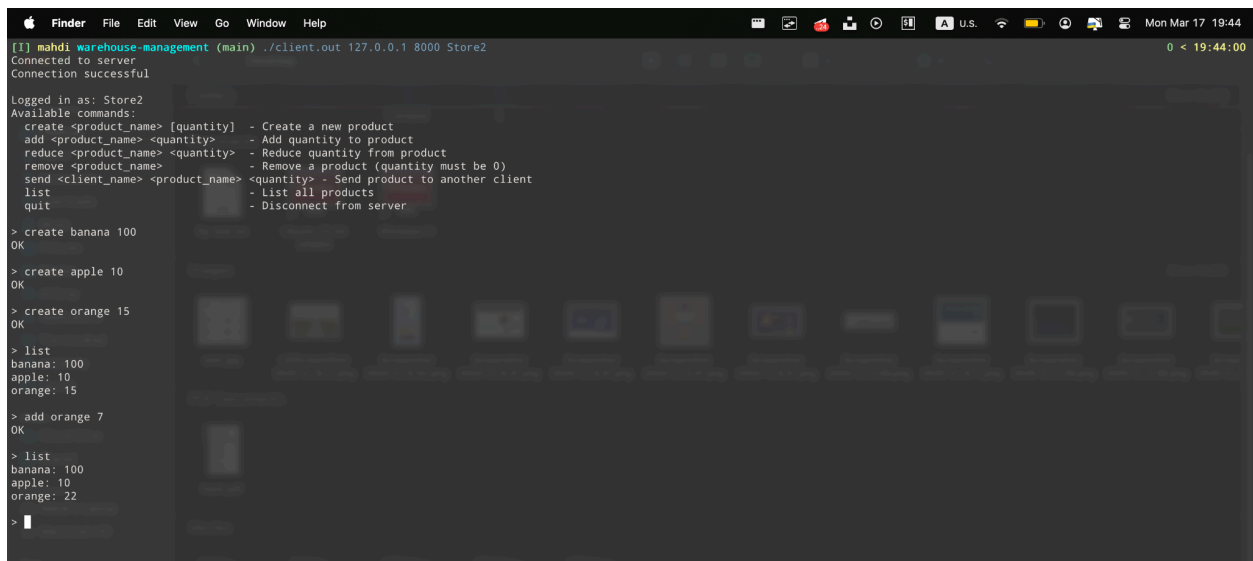
دستورات پشتیبانی شده:

- `list`: نمایش لیست اجناس موجود برای کاربر.
- `[create [new_product_name]`: ایجاد جنس جدید با مقدار اولیه صفر.
- `[create [new_product_name] [initial_count]`: ایجاد جنس جدید با مقدار اولیه مشخص.
- `[add [product_name] [amount]`: افزایش موجودی جنس.
- `[reduce [product_name] [amount]`: کاهش موجودی جنس.
- `[remove [product_name]`: حذف جنس از لیست.
- `[send [target_client_name] [product_name] [amount]`: انتقال جنس به کاربر دیگر.
- `quit`: خروج از برنامه.

نکات پیاده‌سازی:

- استفاده از `pthread` برای مدیریت نخ‌ها.
- استفاده از `mutex` برای جلوگیری از دسترسی همزمان به منابع مشترک.
- بررسی دقیق ورودی‌ها و جلوگیری از خطاها.
- مدیریت حافظه به صورت صحیح.

Client:



```
[1] mahdi:warehouse-management (main) ./client.out 127.0.0.1 8000 Store2
Connected to server
Connection successful

Logged in as: Store2
Available commands:
  create <product_name> [quantity] - Create a new product
  add <product_name> <quantity>    - Add quantity to product
  reduce <product_name> <quantity> - Reduce quantity from product
  remove <product_name> <quantity> - Remove a product (quantity must be 0)
  send <client_name> <product_name> <quantity> - Send product to another client
  list                             - List all products
  quit                             - Disconnect from server

> create banana 100
OK

> create apple 10
OK

> create orange 15
OK

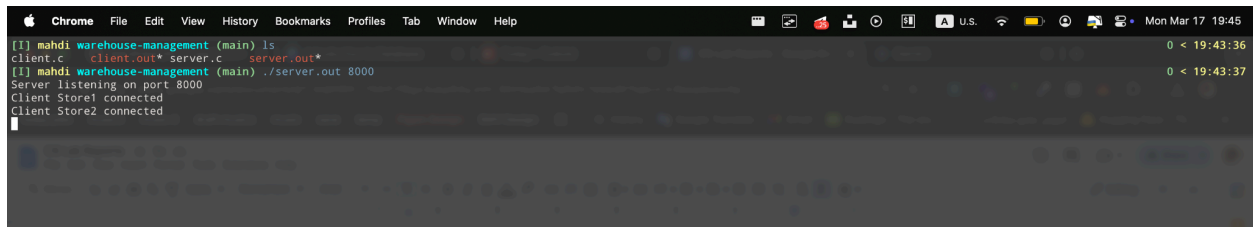
> list
banana: 100
apple: 10
orange: 15

> add orange 7
OK

> list
banana: 100
apple: 10
orange: 22

> 
```

Server:



```
[1] mahdi:warehouse-management (main) ls
client.c  client.out* server.c  server.out*

[1] mahdi:warehouse-management (main) ./server.out 8000
Server listening on port 8000
Client Store1 connected
Client Store2 connected


```