BANK OS

CONTRIBUTORS :

TEAM - 2

SRIPRANAV MANNEPALLI	CS18B036
MAHESWARA REDDY	CS18B032
CHIRAG GUPTA	CS18B006
AVS HRUDAI	CS18B001

Introduction :

The Operating System we developed can be used by the Bank's cashier, manager and also bank account holder. We made it in a way that it covers all the basic tasks required by the banks.

OS concepts that we covered in this project:

- Process Scheduling
- InterProcess Communication.
  - Message Passing.
  - Shared memory.
- Process Synchronization.
- High level memory management.
- Deadlock detection, Deadlock Recovery.
- Device Management.

Features :

Our Operating System has the following features:

- Creating a bank account. We can create as many bank accounts as we need.
- Login to your Bank account.
- Deposit money.
- Withdraw money.
- Transfer money to a different account.
- Show account details..
- Advertisements of the bank.
- Show all accounts details.
- Add review.
- Add comment.

We used ***Screen(vga)*** as a means to show output along with the already provided shell.

Xinu features modified or added.

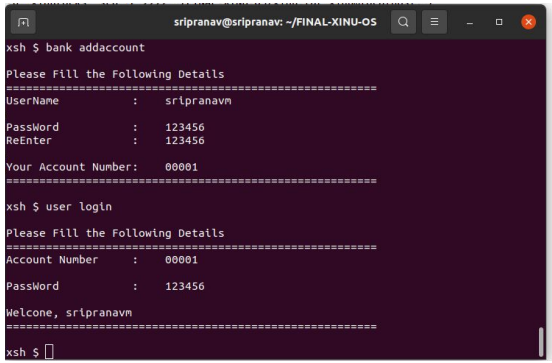
Feature	Modified or Added	Justification
Screen(vga) (A new Device)	Added (device/vga/vga.c)	We added a screen (Output device) to show outputs.The backend UI of VBox is taken as SCREEN.
Deadlock	Added. (system/ deadlock.c)	We needed an algorithm to detect and recover deadlock situations.

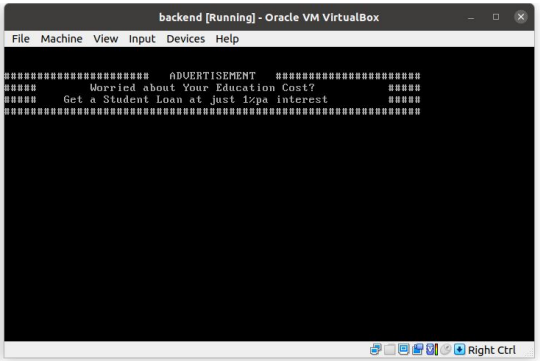
Xinu features used “as-is”.

Feature	Justification
Semaphores : wait , signal	We used semaphores to implement *process synchronization*.
Message Passing :Send , receive , recvclr	We used these to implement *message passing*(IPC).
High Level Memory Management	We used buffer pools to store all the bank details and user details.
Shell commands	We used shell commands as an interaction medium with the OS.

Xinu features Deleted.

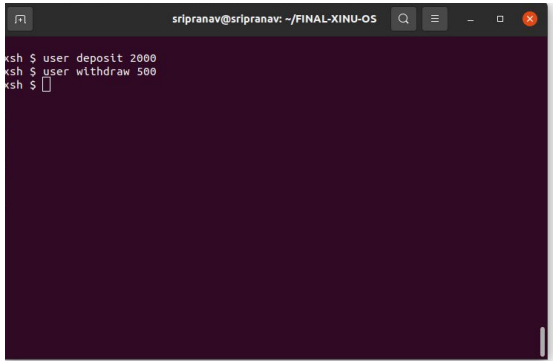
Feature	Justification
Shell commands	We removed all unnecessary shell commands.
device/sdmc	We are not using any SD memory card.
device/rfs	We are not using any remote files.
device/nam	We are not using any of these map function for our features.
device/lfs	We are not using file systems.
Header files in include dir	We removed header files that are not used in our project.

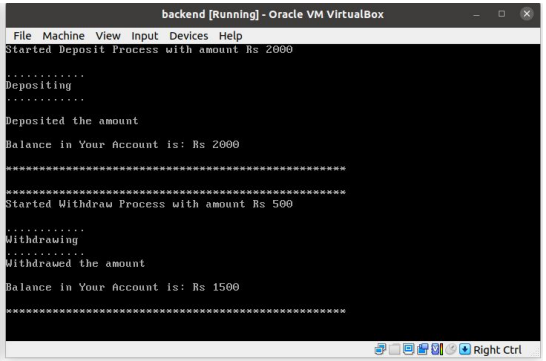




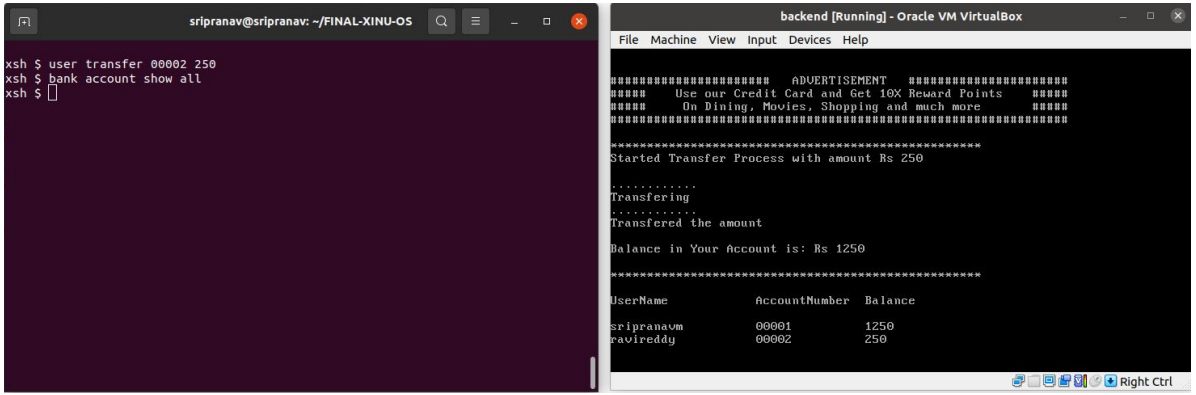
The display on the right is our Screen device(VGA) to show output.

Here we can see account creation and user login and featured advertisements

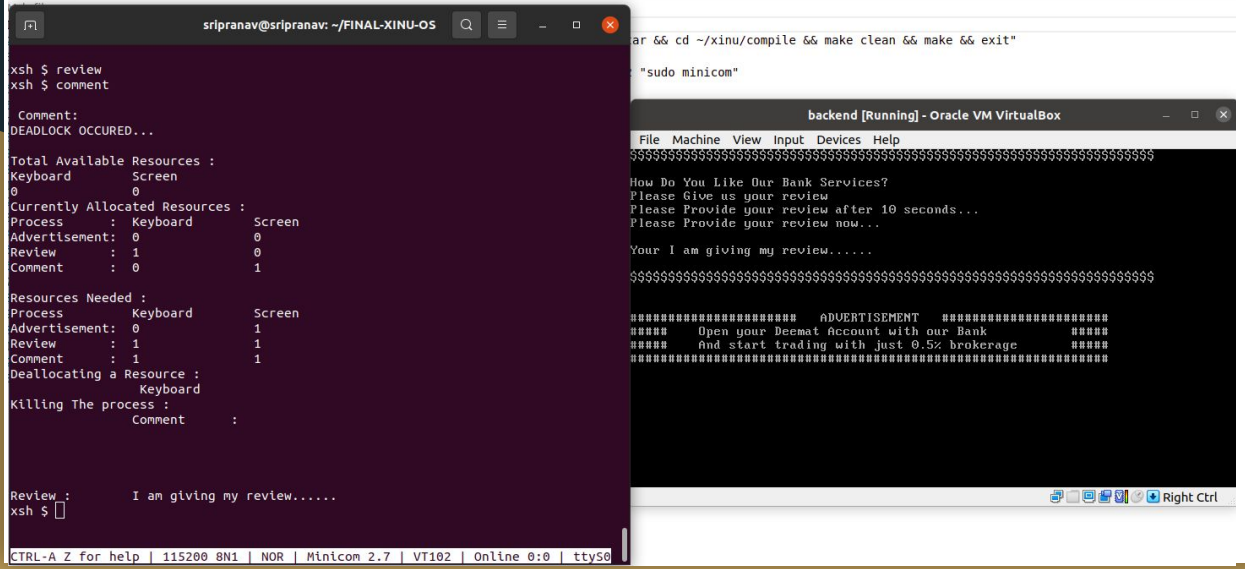




Here we can see user deposits and withdrawals being executed. Each process takes ~10 seconds to execute.



Transfer command being executed here. This command takes ~10 seconds to execute. We can also see all bank accounts available in the bank currently.

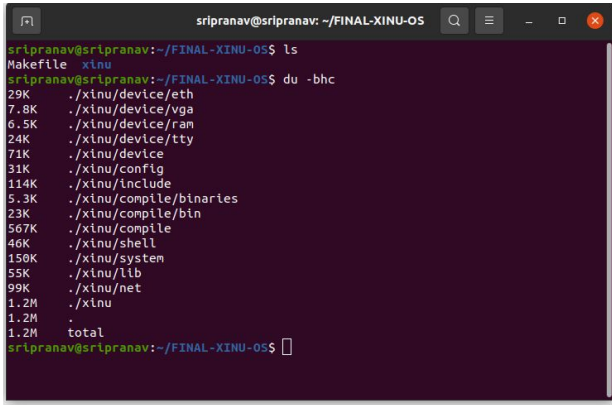


Here we are seeing a *deadlock situation*.
There are two processes comment and review. The resources are keyboard and screen. Both the processes need both the resources. First, the comment process acquires keyboard and review process acquires screen.

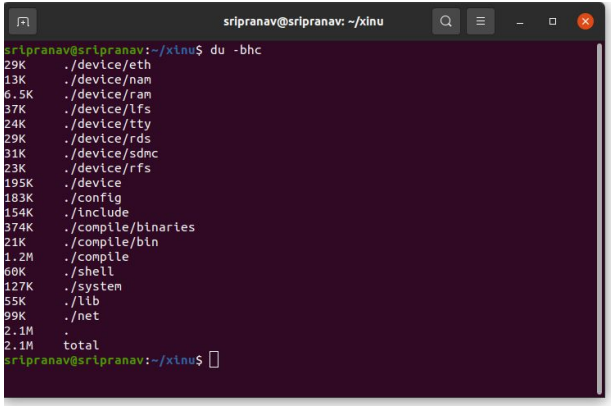
To get out of this deadlock, we kill a process, in this case comment process.

CODE SIZE (BANK OS vs ORIGINAL XINU):

BANK OS
Size : 1.2M



ORIGINAL
XINU OS



CONCLUSION :

We used various OS Features that are present in XINU and we also covered Operating Systems Concepts like Process Synchronization , Process Scheduling , Interprocess Communication (Message Passing,Shared Memory) , Context Switching , High Level Memory Management , Deadlock Detection , Deadlock Recovery.

We also got a practical exposure to the concepts that we learnt during the Operating System Theory course.