



دستور کار جلسه سوم آزمایشگاه سیستم عامل

۱. دستگاهی طراحی کنید که یک بانک کوچک را شبیه سازی کند. ۱۰۰ نفر در این بانک حساب دارند که با اعداد ۰ تا ۹۹ مشخص می شوند. بالانس همه حسابها در ابتدا ۲۰۰۰۰۰۰ واحد پول است. عملیات روی حسابها از طریق نوشتن به فایل درایور انجام می شود و وضعیت حسابها به کمک خواندن از فایل درایور بررسی می شود. فرمت نوشتن به فایل درایور به صورت زیر است:

“Type of transaction,[From],[To],[Ammount]” | “r”
[Type of transaction]: “t” | “d” | “w” (Transfer, Deposit, Withdraw)
[From]: An account number | “-” (0 to 99, indicating a Deposit transaction)
[To]: An account number | “-” (0 to 99, Indicating a withdraw transaction)
[Amount]: A positive integer indicating the amount of money processed during the transaction.

مثال ۱: برای انتقال ۱۲ واحد پول از حساب ۱۷ به حساب ۲۲ باید رشته زیر را در فایل درایور نوشت:

t,17,22,12

مثال ۲: برای واریز ۳ واحد پول به حساب ۴۷ باید رشته زیر را در فایل درایور نوشت:

d,-,47,3

مثال ۳: برای برداشت ۴۴ واحد پول از حساب ۹۷ باید رشته زیر را در فایل درایور نوشت:

w,97,-,44

فرمت رشته خوانده شده از فایل درایور به صورت زیر است:

“[Balance of 0], [Balance of 1], ..., [Balance of 98], [Balance of 99],”

هنگام نوشتن ماژول به نکات زیر توجه کنید:

- در صورت عدم رعایت فرمت نوشتن توسط کاربر باید به کمک printf پیغام مناسبی چاپ نمایید.
- در صورت شکست تراکنش (مثلا کمبود موجودی) باید به کمک printf پیغام مناسبی چاپ نمایید.



یک فایل که شامل چند تابع مورد نیاز برای ایجاد بانک است در اختیارتان قرار گرفته است. از این فایل برای سهولت در پیاده سازی استفاده نمایید.

۲. برنامه‌ای بنویسید که نام دستگاه، نوع تراکنش، حساب مبدأ، حساب مقصد و مقدار پول را از خط فرمان دریافت کند و براساس این مقادیر، تراکنشی با بانک انجام داده و نتیجه را چاپ کند.

۳. شل اسکریپتی بنویسید که کارهای زیر را انجام دهد:

- ماژول طراحی شده را کامپایل کند.
- ماژول کامپایل شده را Load کند.
- به کمک mknod یک فایل درایور به نام iutnode بسازد.
- برنامه شما را اجرا کند.
- فایل درایور را حذف کند.
- فایل‌های خروجی کامپایل شده را پاکسازی کند.