

Лабораторна робота № 1

Виконали: Тивонюк Володимир та Виграновський Марко ФБ-41мн

Тема: „Розгортання систем Ethereum та криптовалют”.

Мета роботи: «Отримання навичок налаштування платформ виконання смартконтрактів та криптовалют»

Для другого типу лабораторних робіт: Провести налаштування обраної системи та виконати тестові операції в системі.

1. Ethereum

Для початку встановимо сам ethereum

```
File Actions Edit View Help
(uvo@kali)-[~/Desktop/new]
$ git clone https://github.com/ethereum/go-ethereum.git
Cloning into 'go-ethereum' ...
remote: Enumerating objects: 138458, done.
remote: Counting objects: 100% (160/160), done.
remote: Compressing objects: 100% (139/139), done.
Receiving objects: 54% (74768/138458), 127.49 MiB | 4.29 MiB/s
```

на калі лінукс чомусь тільки так працює

```
File Actions Edit View Help
(uvo@kali)-[~/Desktop/new/go-ethereum]
$ make geth
go run build/ci.go install ./cmd/geth
go: downloading github.com/cespare/cp v0.1.0
go: downloading github.com/Azure/azure-sdk-for-go/sdk/storage/azblob v1.2.0
go: downloading golang.org/x/crypto v0.35.0
go: downloading github.com/Azure/azure-sdk-for-go/sdk/azcore v1.7.0
go: downloading github.com/Azure/azure-sdk-for-go/sdk/internal v1.3.0
go: downloading golang.org/x/net v0.36.0
go: downloading golang.org/x/text v0.22.0
```

```
(uvo@kali)-[~/Desktop/new/go-ethereum]
$ make all
go run build/ci.go install
>>> /usr/lib/go-1.24/bin/go build -ldflags "--buildid=none -X github.com/ether
ithub.com/ethereum/go-ethereum/internal/version.gitDate=20250325 -extldflags
-trimpath -v -o /home/uvo/Desktop/new/go-ethereum/build/bin/abidump ./cmd/abidu
github.com/ethereum/go-ethereum/signer/fourbyte
github.com/ethereum/go-ethereum/cmd/abidump
>>> /usr/lib/go-1.24/bin/go build -ldflags "--buildid=none -X github.com/ether
```

geth – це основна імплементація Ethereum-клієнта, написана мовою Go. Він дозволяє запускати повноцінний Ethereum-вузол, взаємодіяти зі смарт-контрактами та керувати акаунтами.

```
(uvo@kali)-[~/Desktop/new/go-ethereum]
$ sudo mv build/bin/geth /usr/local/bin/
[sudo] password for uvo:
```

```
(uvo@kali)-[~/Desktop/new/go-ethereum]
$ sudo cp build/bin/clef /usr/local/bin/clef
```

Також нам знадобиться clef – це окремий інструмент для керування акаунтами та підпису транзакцій.

96+Ta9|:0Z;w

```
(uvo@kali)-[~/Desktop/new]
$ clef newaccount --keystore /home/uvo/Desktop/new

WARNING!
Clef is an account management tool. It may, like any software, contain bugs.
Please take care to
- backup your keystore files,
- verify that the keystore(s) can be opened with your password.

Clef is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the GNU General Public License for more details.
Enter 'ok' to proceed:
> ok
```

```
Please enter a password for the new account to be created (attempt 1 of 3)
>

INFO [03-25|13:46:03.399] Your new key was generated          address=0x7e7C194c772dA9192b5d977Ee3F58410c9340770
WARN [03-25|13:46:03.399] Please backup your key file!          path=/home/uvo/Desktop/new/UTC--2025-03-25T17-46-02.049961182Z--7e7c194c772da9192b5d977ee3f58410c9340770
WARN [03-25|13:46:03.399] Please remember your password!
Generated account 0x7e7C194c772dA9192b5d977Ee3F58410c9340770

(uvo@kali)-[~/Desktop/new]
$
```

Запускаємо саме сервіс clef

```
(uvo@kali)-[~/Desktop/new]
$ clef --keystore /home/uvo/Desktop/new --configdir /home/uvo/Desktop/config --chainid 11155111
```

Chain ID — це ціле число, яке визначає мережу Ethereum, до якої підключається клієнт, і для мережі Sepolia воно повинно бути встановлено на 11155111, оскільки Clef використовує його для підпису повідомлень.

```
Enter 'ok' to proceed:
> ok
INFO [03-25|13:51:21.915] Using CLI as UI-channel
INFO [03-25|13:51:22.118] Loaded 4byte database embeds=268,621 locals=0 local=./4byte-custom.json
WARN [03-25|13:51:22.118] Failed to open master, rules disabled err="failed stat on /home/uvo/Desktop/config/masterseed.json: stat /home/uvo/Desktop/config/masterseed.json: no such file or directory"
INFO [03-25|13:51:22.118] Starting signer chainid=11,155,111 keystore=/home/uvo/Desktop/new
light-kdf=false advanced=false
WARN [03-25|13:51:22.158] Failed to start smart card hub, disabling: invalid response code: expected 0, got 21485322
54 (Service was stopped)
INFO [03-25|13:51:22.158] Audit logs configured file=audit.log
INFO [03-25|13:51:22.158] IPC endpoint opened url=/home/uvo/Desktop/config/clef.ipc

----- Signer info -----
* intapi_version : 7.0.1
* extapi_version : 6.1.0
* extapi_http : n/a
* extapi_ipc : /home/uvo/Desktop/config/clef.ipc

----- Available accounts -----
0. 0x7e7C194c772dA9192b5d977Ee3F58410c9340770 at keystore:///home/uvo/Desktop/new/UTC--2025-03-25T17-46-02.049961182Z--7e7c194c772da9192b5d977ee3f58410c9340770

ort=30303 interface=NAT-PMP(192.168.0.1)
WARN [03-25|14:15:54.417] Post-merge network, but no beacon client seen. Please launch one to follow the chain!
```

Для серйозного користування ethereum потрібен консенсусний клієнт (після 2022) - тому в даному випадку будуть лише тестові транзакції.

0x7e7C194c772dA9192b5d977Ee3F58410c9340770

```
(uvo@kali)-[~/Desktop/new]
$ geth --sepolia --datadir /home/uvo/Desktop/new --authrpc.addr localhost --authrpc.port 8551
--authrpc.vhosts localhost --authrpc.jwtsecret /home/uvo/Desktop/new/jwtsecret --http --http.a
pi eth,net --signer=/home/uvo/Desktop/config/clef.ipc --http
INFO [03-25|13:55:19.045] Starting Geth on Sepolia testnet...IT ANY WARRANTY;
INFO [03-25|13:55:19.047] Maximum peer count ETH=50 total=50
INFO [03-25|13:55:19.049] Using external signer details url=/home/uvo/Desktop/config
/clef.ipc
INFO [03-25|13:55:19.051] Set global gas cap cap=50,000,000
INFO [03-25|13:55:19.052] Initializing the KZG library backend=gokzg
INFO [03-25|13:55:19.092] Allocated trie memory caches clean=154.00MiB dirty=256.00
MiB
INFO [03-25|13:55:19.247] Generated JWT secret path=/home/uvo/Desktop/new/j
wtsecret
INFO [03-25|13:55:19.248] HTTP server started endpoint=127.0.0.1:8545 auth
=false prefix= cors= vhosts=localhost with your password.
INFO [03-25|13:55:19.248] WebSocket enabled url=ws://127.0.0.1:8551
INFO [03-25|13:55:19.248] HTTP server started (ful, but WITHOUT ANY endpoint=127.0.0.1:8551 auth
=true prefix= cors=localhost vhosts=localhost or FITNESS FOR A PARTICULAR
WARN [03-25|13:55:19.248] Failed to open wallet details. url=extapi:///home/uvo/Desk
op/conf.. err="operation not supported on external signers"
INFO [03-25|13:55:19.249] Started log indexer
INFO [03-25|13:55:19.262] NAT mapped port proto=TCP extport=30303 intp
ort=30303 interface=NAT-PMP(192.168.0.1)
INFO [03-25|13:55:19.269] NAT mapped port proto=UDP extport=30303 intp
ort=30303 interface=NAT-PMP(192.168.0.1)
INFO [03-25|13:55:19.291] New local node recordes disabled err= seq=1,742,925,319,245 id=313
f05471ddea190 ip=100.79.230.66 udp=30303 tcp=30303 ont no such file or directory
INFO [03-25|13:55:29.277] Looking for peers peercount=1 tried=36 static=
0
INFO [03-25|13:55:39.293] Looking for peers peercount=1 tried=43 static=
0
INFO [03-25|13:55:22.158] Audit logs configured file=audit.log
```

чекає на піри) запускаємо третій термінал і підключаємось

```
(uvo@kali)-[~]
$ geth attach http://127.0.0.1:8545
Welcome to the Geth JavaScript console!

modules: eth:1.0 net:1.0 rpc:1.0

To exit, press ctrl-d or type exit
>
To exit, press ctrl-d or
> eth.accounts
```

отримали запит на clef сервісі - апрув

```
2 7e7c194c772da9192b5d977ee3f58410c9340770
----- List Account request -----
A request has been made to list all accounts.
You can select which accounts the caller can see
[x] 0x7e7c194c772da9192b5d977ee3f58410c9340770
URL: keystore:///home/uvo/Desktop/new/UTC--2025-03-25T17-46-02.04996
770

Request context:
NA → ipc → NA

Additional HTTP header data, provided by the external caller:
User-Agent: ""
Origin: ""
Approve? [y/N]:
>
```

Бачимо наш eth account

```
> eth.accounts
["0x7e7c194c772da9192b5d977ee3f58410c9340770"]
>
> eth.getBalance(eth.accounts[0])
0
```

Статус блоків - синхронізований

```

> eth.syncing
{
  currentBlock: 0,
  healedBytecodeBytes: 0,
  healedBytecodes: 0,
  healedTrieNodeBytes: 0,
  healedTrieNodes: 0,
  healingBytecode: 0,
  healingTrieNodes: 0,
  highestBlock: 0,
  startingBlock: 0,
  syncedAccountBytes: 0,
  syncedAccounts: 0,
  syncedBytecodeBytes: 0,
  syncedBytecodes: 0,
  syncedStorage: 0,
  syncedStorageBytes: 0,
  txIndexFinishedBlocks: 0,
  txIndexRemainingBlocks: 1
}
> eth.blockNumber
0
>

```

```

> eth.chainId
function()
>
> net.peerCount
4

```

Бажано зробити пробну транзакцію - але у даному випадку треба витратити гроші(

Step 4: Get Testnet Ether

In order to make some transactions, the user must fund their account with ether. On Ethereum mainnet, ether can only be obtained in three ways: 1) by receiving it as a reward for mining/validating; 2) receiving it in a transfer from another Ethereum user or contract; 3) receiving it from an exchange, having paid for it with fiat money. On Ethereum testnets, the ether has no real world value so it 4) can be made freely available via faucets. Faucets allow users to request a transfer of testnet ether to their account.

To prevent bots and abuse, this faucet requires a minimum Ethereum mainnet balance of 0.001 ETH on the wallet address being used.

```

..... )^C
> eth.sendTransaction({
.....   from: '0x7e7c194c772da9192b5d977ee3f58410c9340770', to: '0x7e7c194c772da9192b5d977ee3f58410c9340770', value
: web3.toWei(0, 'ether')});
Error: insufficient funds for transfer
    at web3.js:6386:9(39)
    at send (web3.js:5115:62(29))
    at request at <eval>:1:20(13)
>

```

```

allow the chain!
WARN [03-25|14:08:53.950] Served eth_sendTransaction      conn=127.0.0.1:50806 reqid=1
1 duration="166.921µs" err="insufficient funds for transfer"

```

Тому на даному пункті я отримаю лише ворнінг(

--allow-insecure-unlock

Після розблокування аккаунту роблю тестову транзакцію на 0 гривень між собою:

```

To exit, press ctrl-d or type exit
> eth.sendTransaction({
.....  from: eth.accounts[0],
.....  to: eth.accounts[0], // Sending to the same account
.....  value: 0,
.....  gas: 21000,
.....  gasPrice: web3.toWei(5, "gwei")
..... })
"0x52e58db3f0c2332832877b018beeebba8e66884393f4784e6cfb22529355ca0"
>
>

```

```

----- Transaction request -----
to:      0x7e7C194c772dA9192b5d977Ee3F58410c9340770
from:    0x7e7C194c772dA9192b5d977Ee3F58410c9340770 [checksum ok]
value:   0 wei
gas:     0x5208 (21000)
gasprice: 5000000000 wei
nonce:   0x0 (0)
chainid: 0xaa36a7
Request context:
  NA → ipc → NA
Additional HTTP header data, provided by the external caller:
  User-Agent: ""
  Origin: ""
Approve? [y/N]:
> y

```

Отримали транзакцію, але вона довго доходить

```

> eth.getTransaction('0x52e58db3f0c2332832877b018beeebba8e66884393f4784e6cfb22529355ca0');
Error: transaction indexing is in progress
    at web3.js:6386:9(39)
    at send (web3.js:5115:62(29))
    at <eval>:1:19(3)
>

```

Бо сигнатура транзакції то є всередині clef

```
Transaction signed:
{
  "type": "0x0",
  "chainId": "0xaa36a7",
  "nonce": "0x0",
  "to": "0x7e7c194c772da9192b5d977ee3f58410c9340770",
  "gas": "0x5208",
  "gasPrice": "0x12a05f200",
  "maxPriorityFeePerGas": null,
  "maxFeePerGas": null,
  "value": "0x0",
  "input": "0x",
  "v": "0x1546d72",
  "r": "0x8f66867be948ab10b4d7a094faa645e41d3c968bad209ef592325dec0552029f",
  "s": "0x4f186b3a7195edc3fbb27f7426b42b5b1e267e880d5362cf218b7c21db9841d1",
  "hash": "0x52e58db3f0c2332832877b018beeeb8ea8e66884393f4784e6cfb22529355ca0"
}
----- List Account request -----
```

0x52e58db3f0c2332832877b018beeeb8ea8e66884393f4784e6cfb22529355ca0

Висновки:

У ході лабораторної роботи було налаштовано Ethereum-клієнт Geth на Kali Linux, запущено Clef для керування акаунтами та підпису транзакцій. Було отримано та підтверджено запит на підпис транзакції, проте через особливості Clef транзакція затримувалася.