

Note that: \$USERNAME and USERNAME is your surname  
Password: 123

## Simple Linux Navigation Commands

To go home directory

**\$ cd /home/\$USERNAME**

Go into a folder

Change **D**irectory

**\$ cd xyz**

**\$ cd xyz/abc**

Go parent folder

**\$ cd ..**

List folders & files

**\$ ls -al**

Shows where you are

Print **W**orking **D**irectory

**\$ pwd**

Outputs contents of a file

**\$ cat <filename>**

## SSH Connection

Please get your port number of ssh server from student list

**\$ ssh USERNAME@bilgi.merdiven.co -p XXXX**

### **CHANGE PASSWORD**

**\$ passwd**

## Geth Initialize & Geth Attach

To Initialize Ethereum Node & P2P Networking

**\$ geth --datadir /home/\$USERNAME/chain/ --syncmode fast --rinkeby**

To Attach to Ethereum Node & Open JS Console

**\$ geth attach /home/\$USERNAME/chain/geth.ipc**

# Account

List accounts

> **eth.accounts**

Get account address

> **eth.accounts[0]**

Create new account (**DONT FORGET YOUR PASSPHRASE !!!**)

> **web3.personal.newAccount()**

Set default account !!!

> **eth.defaultAccount = eth.accounts[0]**

Unlock account

> **web3.personal.unlockAccount(eth.accounts[0],PASSPHRASE,999999)**

Then enter your passphrase

List encrypted key files

**\$ ls /home/\$USERNAME/chain/keystore**

**Homework: Copy and save your key file**

> **cat /home/\$USERNAME/chain/keystore/UTC....**

**Send 0.01 ether**

**to 0xc02477b166dd7d807ba24895d5482dcc47a4437c**

**from MyEtherWallet.com @Rinkeby Network.**

Get Balance

> **eth.getBalance(eth.accounts[0])**

# Transactions

Denominations

1: wei      10<sup>12</sup>: szabo      10<sup>15</sup>: finney      10<sup>18</sup>: ether

```
> receiver = "0xc02477b166dd7d807ba24895d5482dcc47a4437c"
> sender = eth.accounts[0] which is your account
> amount = web3.toWei(0.02,"ether")
> dat = web3.fromUtf8("xyz abc arbitrary data")
> txnData = {from:sender, to:receiver, value:amount,data:dat}
> eth.sendTransaction(txnData)
```

results some hash like:

0x15a7fb90c84c97cba0652330fea.....

```
> txnHash = "0x15a7fb90c84c97cba0652330fea..."
> eth.getTransaction(txnHash)
```

**gas** : Allocated gas amount for that transaction

**gasPrice** : Price in Wei for each unit of gas

*gas\*gasPrice* : Total amount in Wei allocated for that transaction

**input**: arbitrary data *web3.fromUtf8("xyz abc arbitrary data") (What is UTF8)*

**blockNumber**: which block the txn is involved

**blockHash**: hash of that block

```
> eth.getTransactionReceipt(txnHash)
```

**gasUsed** : gas used to process that transaction by miner. Helps to check if enough *gas\*gasPrice* send. if it is equal to **gas** parameter, probably transaction went wrong.

```
> eth.getBlock(100456)
```

or

```
> eth.getBlockByHash("0x2bdf0d7f9cef27c7ee2a8e5ad04b..")
```

**gasLimit** : It limits number of transactions in other words processing capacity in a block

**Research Homework: Why such limit is needed?**

**transactions** : Processed and validated transactions in that block

**timestamp**: is unix timestamp

**parentHash**: hash of previous block

# Contracts & Solidity

Default contract directory on your host machine:

`/home/$USERNAME/shared/contracts/`

Default script directory on your host machine:

`/home/$USERNAME/shared/scripts/`

Note:

Contract is not a *contract* actually.

Contract is an object.

## SimpleContract

```
contract SimpleStorage(){

    uint value;
    address creator;

    function SimpleStorage(){
        // default value is 1000
        value = 1000;
        creator = msg.sender;
    }

    function setValue(uint _value){
        value = _value;
    }

    function getValue() constant returns(uint){
        return value;
    }

    function getCreator() constant returns(address){
        return creator;
    }
}
```

## **Compile a contract:**

```
$ solc --abi --bin /home/$USERNAME/shared/contracts/SimpleStorage.sol
```

Output1:

Application Binary Interface

Output2:

Binary data

Both are required to deploy a contract onto ethereum virtual machine.

## **Deploy a contract:**

```
> bin = "0x6060604052341561000f5760....."
```

```
> abi = [{"constant":true,"inputs":[],"name":"getValue","outputs":
[{"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","type":"functi
on"}, {"constant":false,"inputs":
[{"name":"_value","type":"uint256"}],"name":"setValue","outputs":
[], "payable":false,"stateMutability":"nonpayable","type":"function"}, {"inputs":
[], "payable":false,"stateMutability":"nonpayable","type":"constructor"}]
```

```
> loadScript("/home/$USERNAME/shared/scripts/deploy.js")
```

```
> simpleContract = deploy(bin,abi)
```

```
> simpleContract.setValue(99)
```

```
> simpleContract.getValue()
```