Borbély Katalin - CDLEBS

Szigeti Soma - BQHM70

S15\_2

# Feladatkiírás

HW15: Virtual server rental

**Problemstatement**

Implement a smartcontractforvirtual server rental.

* The contractshouldhaveexactlyoneowner (the server's owner).
* Clientscanbuyaccesstovirtual servers fortimeslots of 30 minutes.
* The ownersetstheprice per timeslot.
* Access totimeslotscan be transferredbetweenclients.
* Anyonecancheckif a client has accesstothemachineatanygiventime.
* The ownercanwithdrawtheprofits.

**Examplescenario**

Alice owns a server park. Shecreates an instance of thiscontracttomanageaccesstoone of her servers, settingtheprice per timeslotto 0.01 ether.

Bob reservesthetimeslotbetween 11:00am and 11:30am this Sunday usingthetimeslot's numbersincecontractcreation. Later, he decidestotransferhisreservationtoClaire. After 11:00am, Clairesendsheraddress and a packetsignedwithherprivatekeytothe server. The server validatesthepacket, checks is she has accessduringthecurrenttimeslot, and grantsaccesstoClaire.

**Contractinterface**

* constructor: functionReserveVS(uint256 pricePerSlot)
  + The creator of thecontract (msg.sender) is theowner.
  + pricePerSlot is thepriceforrentingthe server for a singletimeslot.
* reserve: functionreserve(uint24 slotId) payable
  + ReservetheslotId'th timeslotsincecontractcreation. Forexample, ifthecontractwascreatedtodayat 00:00, thenslotnumber 16 would be theonefrom 07:30 to 08:00, whileslotnumber 64 would be thesametimeslotonthenextday.
  + Thistransactionsucceedsonlyifthetimeslot has notyetbeenreserved, and thesendertransfersthecorrectamount of ether.
* transfer: functiontransfer(uint24 slotId, addressto)
  + Transferownership of theslotId'th timeslottoaddressto.
  + Succeedsonlyifthegiventimeslotoriginallybelongedtothesender of thetransaction (msg.sender).
* validate: functionvalidate(addressclient) viewreturns (bool)
  + Returnstrueifclient is entitledtousethemachineatthetime of thevalidation.
  + Can be calledbyanyone. (Inourusecase, thiswill be calledbythe server itself.)
* withdraw: functionwithdraw()
  + Transfer profit tothesender.
  + Onlytheowner of thecontractcancallit.

Contractskeleton

pragmasolidity ^0.4.21;

contractReserveVS{

// <contract\_variables>

// </contract\_variables>

functionReserveVS(uint256 pricePerSlot) public {

// TODO

}

functionreserve(uint24 slotId) publicpayable {

// TODO

}

functiontransfer(uint24 slotId, addressto) public {

// TODO

}

functionvalidate(addressclient) publicviewreturns (bool) {

// TODO

}

functionwithdraw() public {

// TODO

}

}

# Okos szerződés funkciók

Egy szerverpark megvalósítása a cél. A szerverparkhoz tartozik egy gazda, akitől lehet bérelni félórás időtartamokban a szervereket. Egy időpontban csak egy szerver bérelhető. Egy szervert egyszerre csak egy ember tud bérelni/használni.

A szerver bérlésének feltétele: az adott időpontra ne legyen még foglalás. A bérlő a foglalás időpontjában rendelkezzen a foglaláshoz szükséges etherrel.

Egy már lefoglalt szerver átadható egy másik embernek. Csak olyan időpont adható át, melyet a bérlő foglalt le. A folyamatot a bérlő kezdeményezi. A lefoglalt szerver használati jogának átadásakor is az eredeti bérlőtől kerül levonásra az ether.

A szervert csak foglalás időtartalma alatt használhatja a bérlő és csak a bérlő.

A foglalás pillanatában a bérlés költsége zárolásra kerül a bérlő számláján. Melynek levonását és a saját számláján való jóváírását később a szerverpark gazdája kezdeményezheti.

**Konstruktor: function ReserveVS(uint256 pricePerSlot) public**

A konstruktorban létrehozunk egy szerverparkot. A szerverpark gazdája a hívó (msg.sender). Szerverpark létrehozásához szükséges még megadni egy szerver egy időtartamra való bérlésének az árát. A megadott árat átalakítjuk wei-ből etherre.

**Foglalás: function reserve(uint24 slotId) public payable**

Foglalás csak a szerverpark létrehozása után történhet. A foglalónak meg kell adnia a slotId-t azaz, hogy a szerverpark létrehozása óta melyik blokkban szeretne szervert bérelni. Minden blokk egy félórás idő intervallumot takar. Sikeres a foglalás, ha az adott intervallumot még nem foglalta le senki, illetve a bérlő rendelkezik a megfelelő mennyiségű etherrel. Egyszerre csak egy intervallumot lehet lefoglalni. Amennyiben több intervallumot szeretne a bérlő lefoglalni, úgy többször kell foglalást kezdeményeznie.

Sikertelen foglalás esetén hibaüzenettel kerül értesítésre a foglaló.

Sikeres foglalás esetén levonásra kerül a bérlési díj (átküldött ether) és lefoglalásra kerül az intervallum a bérlőnek.

**Hely átadás: function transfer(uint24 slotId, address to) public**

A bérlő átadhatja az általa lefoglalt időpontot egy másik embernek. Ehhez szükséges megadni az intervallumot és az új bérlő ID-ját. Egyszerre csak egy időpont adható át más embernek. Sikeres az átadás, ha a megadott idő intervallum az eredeti bérlőé. Ekkor az adott intervallum bérlőjét kicseréljük az új bérlőre. Hely átadásnál az új bérlőnek nem kell fizetnie a szerver bérlésért.

Sikertelen átadás esetén hibaüzenettel értesítjük a bérlőt.

**Validálás belépéshez: function validate(address client) public view returns (bool)**

Ellenőrzésre kerül, hogy az adott kliens hozzá tud a férni az adott időpillanatban a szerverhez.

Sikeres hozzáférés akkor történik, ha a lefoglalt intervallumba próbál belépni a bérlője. Amennyiben nem a bérlő próbál hozzáférni a szerverhez, vagy a bérlő, de rossz idő intervallumban úgy a szerverhez való hozzáférés megtagadásra kerül.

**Ellenérték felvétele: function withdraw()**

A foglalási díjat a szerverpark gazdája beszedi.

# Specifikációnak való megfelelés

1. A kontraknak egy tulajdonosa lehet a szerverpark tulajdonosa – a **function ReserveVS(uint256 pricePerSlot) public** függvény „owner = msg.sender” sorával állítjuk be a tulajdonost.
2. A bérlő félórás időintervallumokat tud bérelni – a **functionreserve(uint24 slotId) payable** függvény valósítja meg a bérlést. Sikeres bérlés esetén a megadott intervallumhoz hozzárendelésre kerül a bérlő, ezt a függvény „reservation[slotId] = msg.sender” része valósítja meg.
3. A tulajdonos meghatározhatja az időintervallum bérlésének árát – a **function ReserveVS(uint256 pricePerSlot) public** függvény „price = pricePerSlot \* (10 \*\* 18)” sora valósítja meg ezt a követelményt.
4. A lefoglalt idő átadható egy másik embernek – a function **transfer(uint24 slotId, address to) public** függvény implementálja az átadást. A „reservation[slotId] = to” műveletben történik meg konkrétan a szerver bérlésének átírása az új tulajdonosra.
5. Bárki bármikor ellenőrizheti, hogy a kliensnek van e hozzáférése a géphez – a **function validate(address client) public view returns (bool)** függvény valósítja meg. A „uint diff = 1 + (block.timestamp - VSCreationDate) / 60 / 30” sor kiszámolja, hogy melyik időintervallumban vagyunk – amikor foglalt az az első – a “return reservation[diff] == client’ pedig visszaadja hogy hozzáférhet e az időintervallumban a szerverhez a kliens vagy sem.
6. A tulajdonos elveheti a foglalásokért járó összeget – a **function withdraw() public** függvény valósítja meg az átvételt. Ennek is konkrétan a „owner.transfer(balanceToBeWithDrawn)” sora.

# Teszt szcenárió futtatása

**Szerverpark létrehozása:**

ReserveVS függvény meghívása a következő cím által: „0xca35b7d915458ef540ade6068dfe2f44e8fa733c”. Paraméterként 25-öt adok át a függvénynek, mely a 25 ether-es szerverfoglalási díjat reprezentálja egy időintervallumra.

**Szerver lefoglalása**:

A reserve függvény meghívása a következő cím által „0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db”. Paraméternek átadom a lefoglalni kívánt időintervallumot, ami legyen az első. Illetve a 3. időintervallumot lefoglalta egy másik cím, a “0x583031d1113ad414f02576bd6afabfb302140225”. Mindketten 25 ehtert utaltak át, melyet a value mezőben adtam meg. A szerver lefoglalása után a szerverfoglalási díj levonásra került a bérlőktől.

**Szerver lefoglalása kevesebb átküldött etherrel:**

A reserve függvény meghívásra került a “0x583031d1113ad414f02576bd6afabfb302140225” által. Paraméterként átadásra került egy 5-ös szám, mely azt reprezentálja, hogy a bérlő az 5. időintervallumban szeretne lefoglalni egy szervert. Kevesebb ethert küldök át, a szükséges 25 helyett csak 10-et. A szerverfoglalás nem történik meg, a művelet hibára fut.

**Szerver lefoglalása elküldött ether nélkül:**

A reserve függvény meghívásra került a “0x583031d1113ad414f02576bd6afabfb302140225” által. Paraméterként átadásra került egy 6-os szám, mely azt reprezentálja, hogy a bérlő az 6. időintervallumban szeretne lefoglalni egy szervert. Nem küldök ethert. A szerverfoglalás nem történik meg, a művelet hibára fut.

**Lefoglalt szerver bérleti jogának áthelyezése**:

A “0x583031d1113ad414f02576bd6afabfb302140225” által lefoglalt 3. időintervallumbeli szerver bérlési jogosultságát átadom a “0x14723a09acff6d2a60dcdf7aa4aff308fddc160c”-nek. Ehhez a transfer függvényt hívom meg, melynek paraméterül a következőket adom meg:

3, “0x14723a09acff6d2a60dcdf7aa4aff308fddc160c”

Azaz az időintervallumot és az új bérlőt. A bérleti jog átadása sikeresen megtörténik.

**Lefoglalt szerver bérleti jogának áthelyezése, rossz időintervallum megadásával**

A transfer függvényt meghívom a 2, “0x583031d1113ad414f02576bd6afabfb302140225” értékkel. Az adott bérlő nem a 2-es időintervallumot foglalta le, így a jogosultság átadása nem történik meg. Hibaüzenetet kapunk.

**Lefoglalt szerver bérleti jogának áthelyezése, nem a megfelelő bérlő hívja meg a függvényt**

A transfer függvényt meghívom az 1, „“0x14723a09acff6d2a60dcdf7aa4aff308fddc160c” parameter átadásával a “0x583031d1113ad414f02576bd6afabfb302140225” címről. Az adott időintervallumon lefoglalt szervernek nem a függvény hívását elindító a bérlője. A jogosultság átadása nem történik meg, hibaüzenetet kapunk.

**Szerver használhatóságának ellenőrzése:**

A validate függvényt meghívjuk. Paraméterként átadjuk neki a következő címet:

„0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db”. A függvény igaz értéket ad vissza, mivel a megadott bérlőhöz tartozik az adott időintervallum, amikor a függvény meghívásra került.

**Szerver használhatóságának ellenőrzése, nem a bérlő címét adjuk meg paraméterként:**

A validate függvényt meghívjuk. Paraméterként átadjuk neki a következő címet:

„0x583031d1113ad414f02576bd6afabfb302140225”. A függvény hamis értéket ad vissza, mivel a nem amegadott bérlőhöz tartozik az adott időintervallum, amikor a függvény meghívásra került.

**Bérleti díj beszedése:**

Meghívjuk a withdraw függvényt a „0xca35b7d915458ef540ade6068dfe2f44e8fa733c” címről. A megadott cím a tulajdonosé. A lefoglalt szerverekért fizetett 50 ether megjelenik a tulajdonos számláját.

**Bérleti díj beszedése, nem a tulaj hívja meg a függvényt:**

Mghívjuk a withdraw függvényt a „0xdd870fa1b7c4700f2bd7f44238821c26f7392148” címről. Az adott cím nem a tulajdonoshoz tartozik, így hibát kapunk. A lefoglalt szerverekért járó díj nem kerül átutalásra senkinek sem.