

ClusterRack dokumentáció



Ez a dokumentum a ClusterRack klaszter-kezelő projekt dokumentációját tartalmazza.

Három fő részre tagolódik:

- Bevezetés - (a feladat, programunk leírása, csapatunk ismertetése)
- Felhasználói dokumentáció - (feature készlet, a vizuális felhasználói felület, a konzol)
- Fejlesztői dokumentáció - (tech stack, együttműködés, fájlok, program technikai működése)

Bevezetés

A programunk különböző klaszterek kezelésére készült. A felhasználó képes egyszerre több klasztert is kezelni, ezeknek nevét, gépeit, programjait változtatni. Emellett az applikáció több eltérő klaszter között gép és program mozgásokat is véghez tud vinni. A terméknek van saját parancsoros kezelőfelületi (CLI) módja a szakemberek számára.

Ajánlott gépi követelmények:

- Operációs rendszer: Windows 10 vagy Windows 11
- Python: 3.13.2-es verziója
- A könyvtárakat és azok verzióját megtalálja a **requirements.txt** fájlban, a `pip install -r ./requirements.txt` parancssal pedig azonnal és egyszerűen fel tudja telepíteni azokat.

Tesztelési állományokat biztosít a program, melyek a programon belüli **test_files** mappában helyezkednek el.

Az eredeti feladat

A csapatoknak egy olyan programot kellett készítenie melynek tudnia kellett számontartani egy klaszter állapotát, valamint meg kellett tudnia változtatnia az alatta lévő gépeket és programokat az utóbbi példányaival együtt.

Csapatunk

Csapatunk az **undefined** csapat a debreceni *DSZC Mechwart András Gépipari és Informatikai Technikum* 10.F tagjai.



- Kovács Dániel Benedek:
 - Szándékomban áll jól működő és kinézettel rendelkező programokat kiadni a kezemből. A csapatban eddig mindig az én feladatom volt a UI megtervezése és kivitelezése.
 - Személyes projektjeim miatt már foglalkoztam kezelőfelületek elkészítésével python-ban, így most is szívesen magamra vállaltam.
- Borók Máté
 - A program CLI felületét és az abban végrehajtható parancsokat készítettem el.
 - Ez egy teljesen új terület volt számomra, úgy döntöttem, hogy megpróbálom a legtöbb dolgot magamtól csinálni külső segítség nélkül.
 - Programozás mellett még dobolok, sokat játszok és sportolok.
- Tóth Péter Sándor
 - Szeretek kreatív munkákon dolgozni és komplex feladatokat megoldani, szabadidőmben játékokat fejleszték, Kung-Fu-zok, fagottozok és videójátékokkal játszok
 - A csapatban a program háttérében futó backend részt készítettem el..

Felhasználói dokumentáció

Alap funkciók

- **A klaszter kezelése:** A programnak figyelnie és módosítania kell a klaszteren futó folyamatokat.
- **Fájlok és mappák használata:** A számítógépeket mappák, a folyamatokat fájlok jelképezik.
- **Erőforrások figyelése:** A számítógépek processzor- és memória használatát nyilván kell tartani.
- **Indítás és ellenőrzés:** A rendszer indulásakor megvizsgálja, hogy minden programpéldány rendben fut-e.
- **Monitoring:** A felhasználó lekérheti az aktuális klaszterállapotot.
- **Számítógépek kezelése:** Lehet gépeket hozzáadni vagy törölni.
- **Programok indítása és leállítása:** Csak a megfelelő erőforrások mellett indíthatók új folyamatok.
- **Bemenet ellenőrzése:** A programnak nem szabad összeomlania rossz bemenet esetén.

Bővített funkciók

- **Tisztítási és fájl validálási rendszer :** Az ismeretlen fájlokat és mappákat a felhasználó engedélyével tölti vagy ignorálja a program
- **Gyökér szintű hozzáférés :** A rendszer képes számon tartani több klasztert egy fájlban és műveleteket végrehajtani közöttük
- **Egyedi parancsvégrehajtó (Custom command interpreter) :** A konzolunk része, ami egy több kényelmi funkcióval ellátott parancsértelmező rendszer. Tartalmaz többek között autocompletet és egyedi batch fájlból való parancsbeolvasási funkciókat is.
- **Konzol felület (Command Line Interface) :** Általunk készített konzol, amely kihasználja a parancsvégrehajtó erejét. Ezen keresztül a tapasztalt felhasználók gyorsabban tudják kezelni problémáikat..
- **Vizuális felhasználói felület :** A vizuálist felhasználói felületen keresztül könnyű és gyors hozzáférést biztosítunk az applikáció funkcióihoz összes felhasználónk számára
- **Három különböző teherelosztási algoritmus:** Többféle mód a klaszteren való programok automatikus kiosztására.

A feature készletről való pontos leírást a fejlesztői dokumentumokban találja.

A program működtetése vizuális felhasználói felületen keresztül

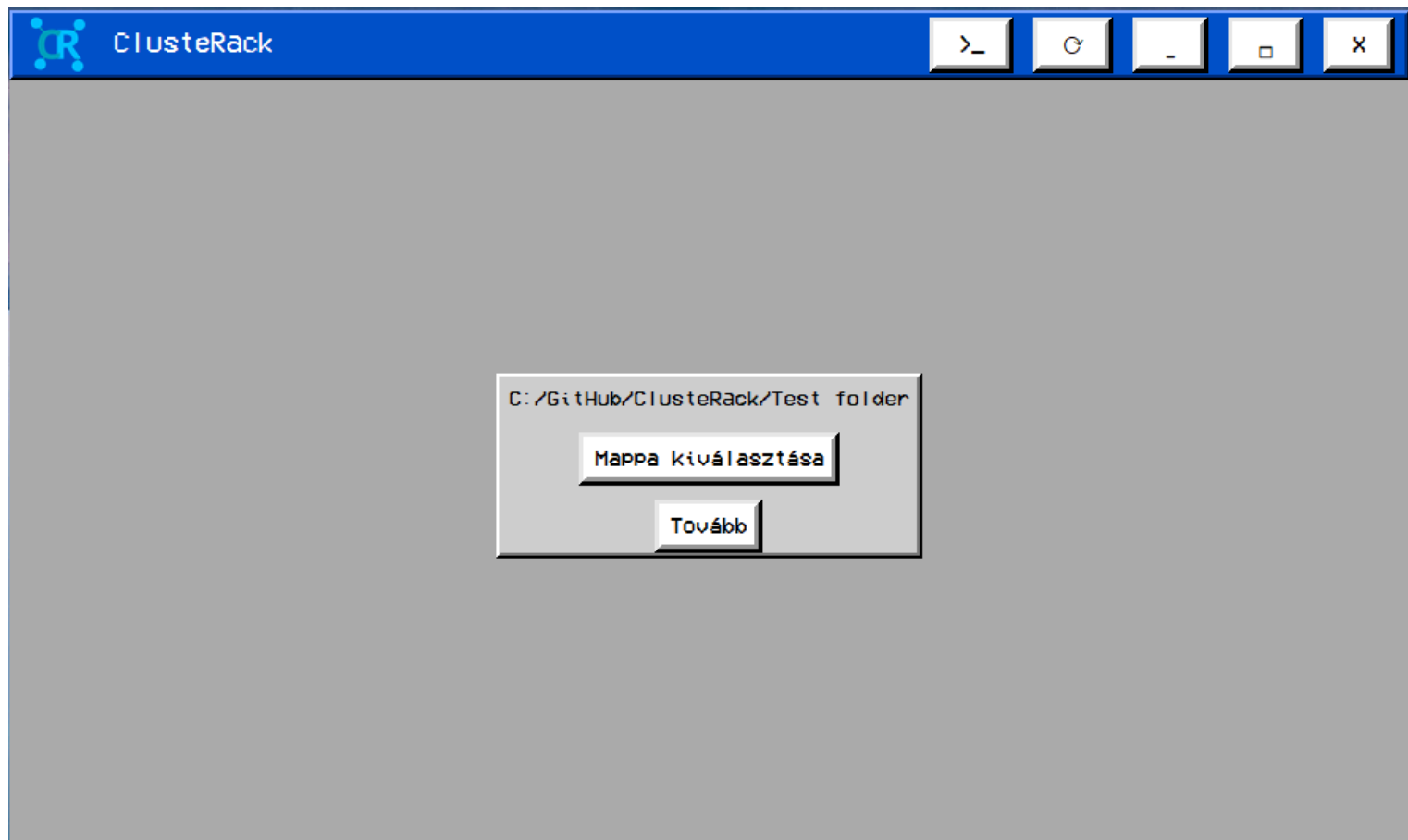
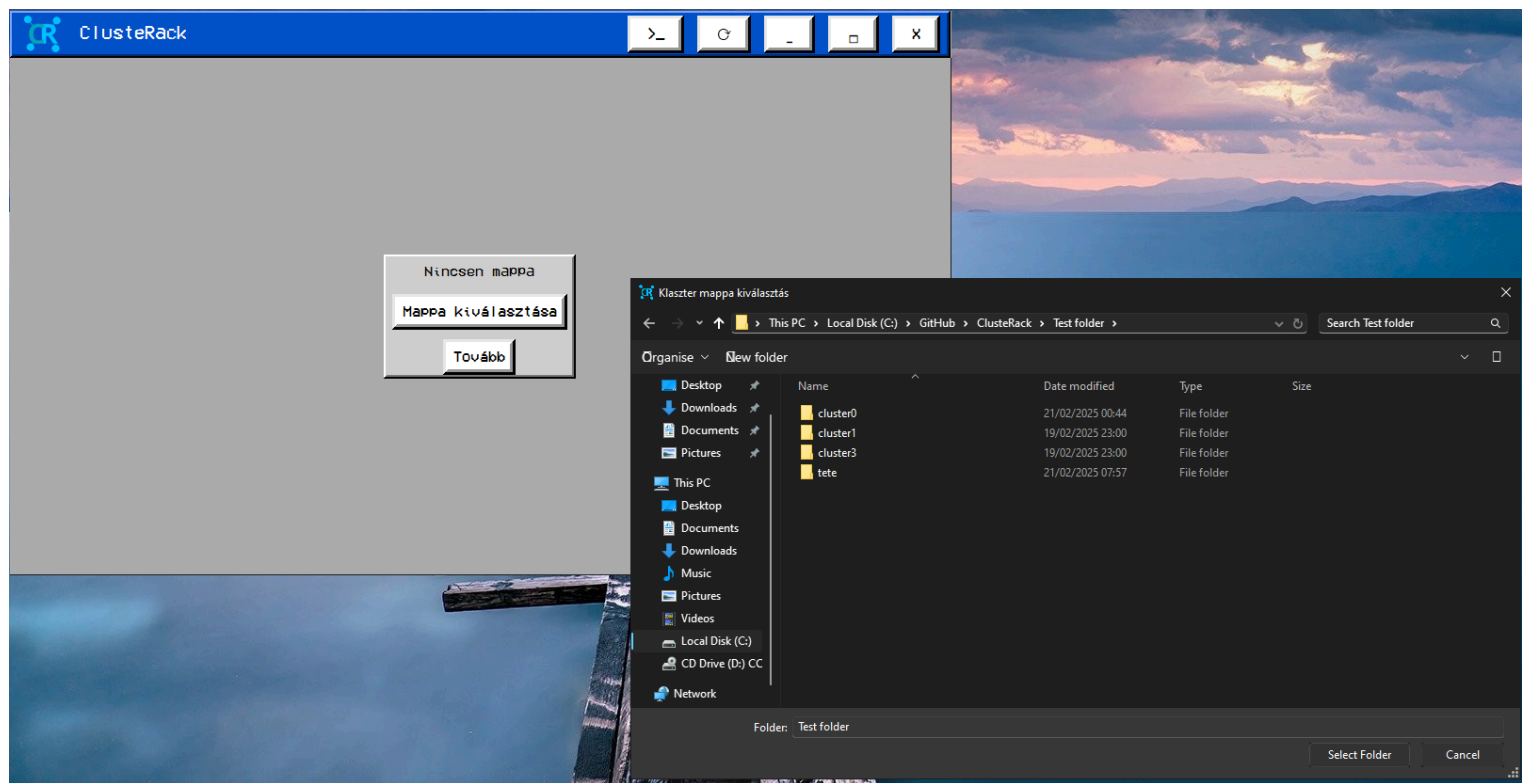
Program indítása

Először a gyökeret (root) kell kiválasztani. Ennek kell tartalmaznia a klasztereket. A gyökér mappa tallózása után le tudja még ellenőrizni a megadott útvonalat. Amennyiben a megfelelő mappát választotta ki, kattintson a "tovább" gombra.

FONTOS: Minden funkció végrehajlásánál futtatunk hiba kereséseket. A hibák egy külön felugró ablakban jelennek meg. A dokumentáció nem fog kitérni minden hibalehetőségre, csak a program rendeltetésszerű használatát mutatja be.

FONTOS: Az olyan felugró ablakoknál, ahol az opciók be vannak számozva mindig csak az általa választott opció számát kell megadni a mezőben.

VIGYÁZAT: A gyökérmappa inicializálásakor lefut egy tisztítási folyamat, ami törli a hibát okozható mappákat és/vagy fájlokat.



KIEGÉSZÍTÉS: Ha a főmappa tartalmaz hibás klasztereket, számítógépeket és/vagy ismeretlen fájlokat, akkor figyelmeztet a hibákról és el tudja dönteni, hogy mit szeretne csinálni.



ClusterRack Subwindow

Ismeretlen fájl a efefe számítógépen: tezt-5I9tnT copy

1: Törlés

2: Megtartás (Figyelmeztetés: Lehetséges hogy destabilizálja a számítógépet)

Írja be választását(1/2):



ClusterRack Subwindow

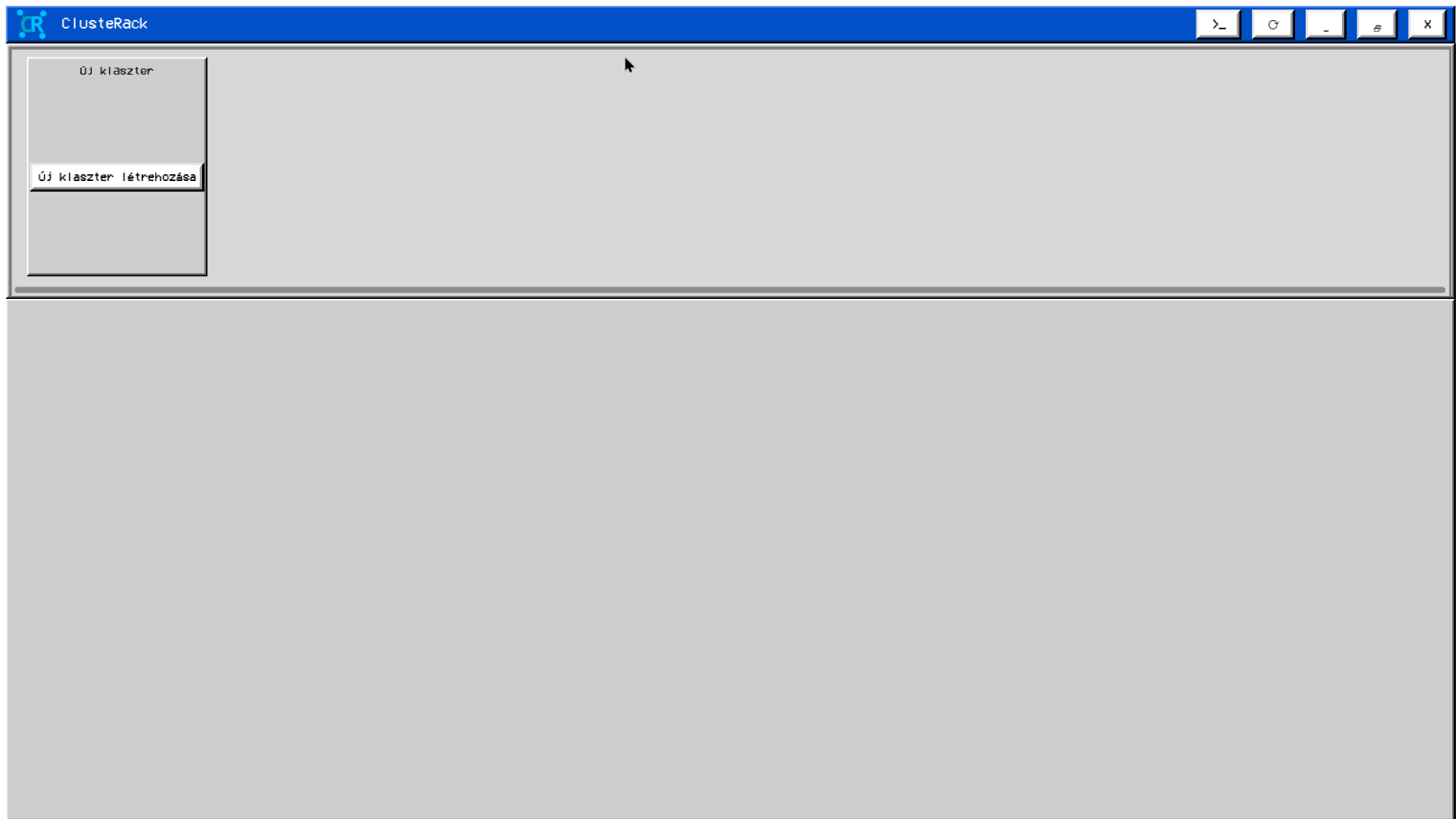
Nincs konfigurációs fájl a C:\GitHub\ClusterRack\Test folder\teszt_klaszter\9ep1 számítógépnek!

Szeretne generálni egyet?

1 - Igen

2 - Nem

Ezek után a program betölti azt a felületet, amiről tudja kezelni a klasztereket.



Gombok

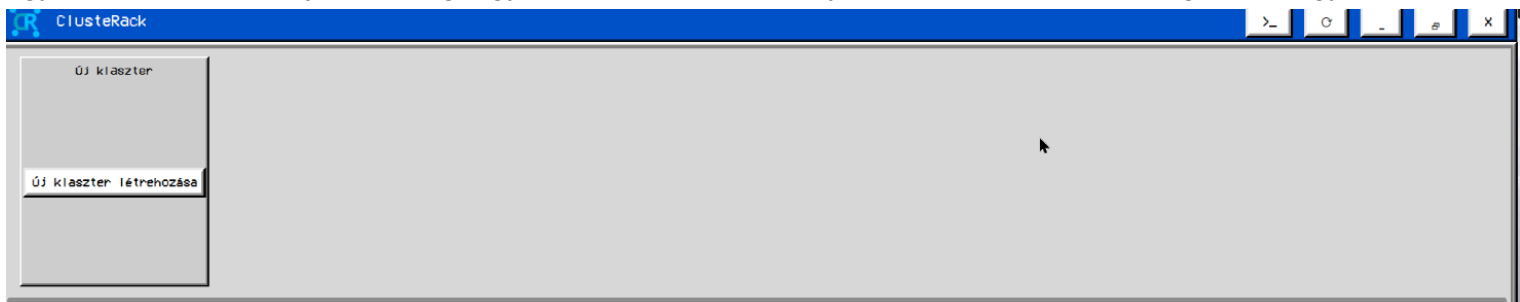


A gombok működése balról-jobbra:

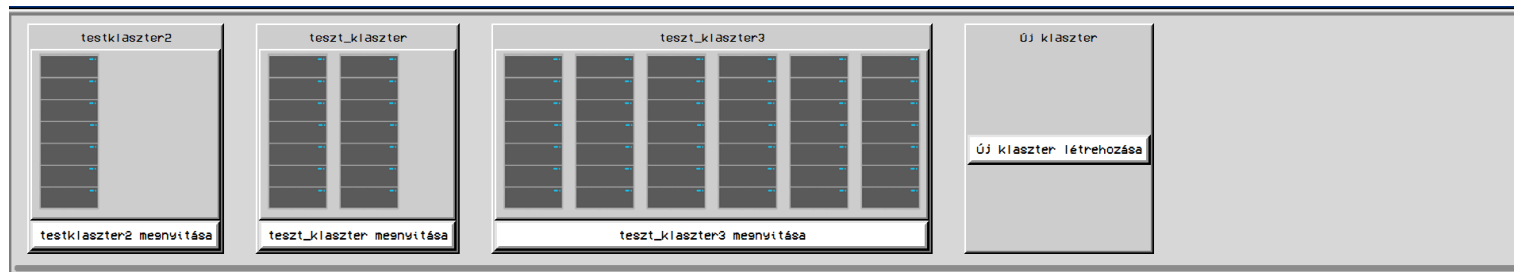
1. **Parancssor:** megnyitja a program parancssorát, ahol ugyanabban a mappában dolgozhatunk a beépített parancsokkal, csak egy CLI-t használva.
2. **Újratöltés:** a program nem frissül, ha kívülről módosítjuk a fájlokat. Ezzel a gombbal újratölti magát a program és megjelennek a változtatások.
3. **Minimalizálás:** leteszi a programot a tálcára
4. **Maximalizálás:** kiteszi maximális képernyőre a programot
5. **Bezárás:** kilép a programból

Klaszter Lista

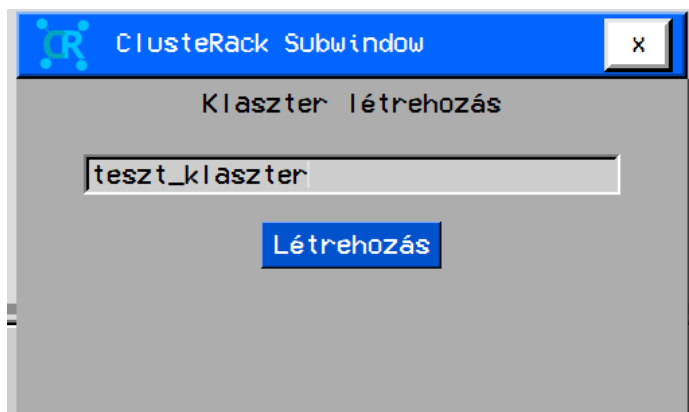
Egy vízszintes listában jeleníti meg a gyökeri klasztereit. Amennyiben nincsenek, lehetőséget ad, hogy készítsen.



KIEGÉSZÍTÉS: A klaszter lista egyaránt megjeleníti, hogy mennyi gépet tartalmaz egy klaszter. A gépeket rack szekrényekkel ábrázolja,



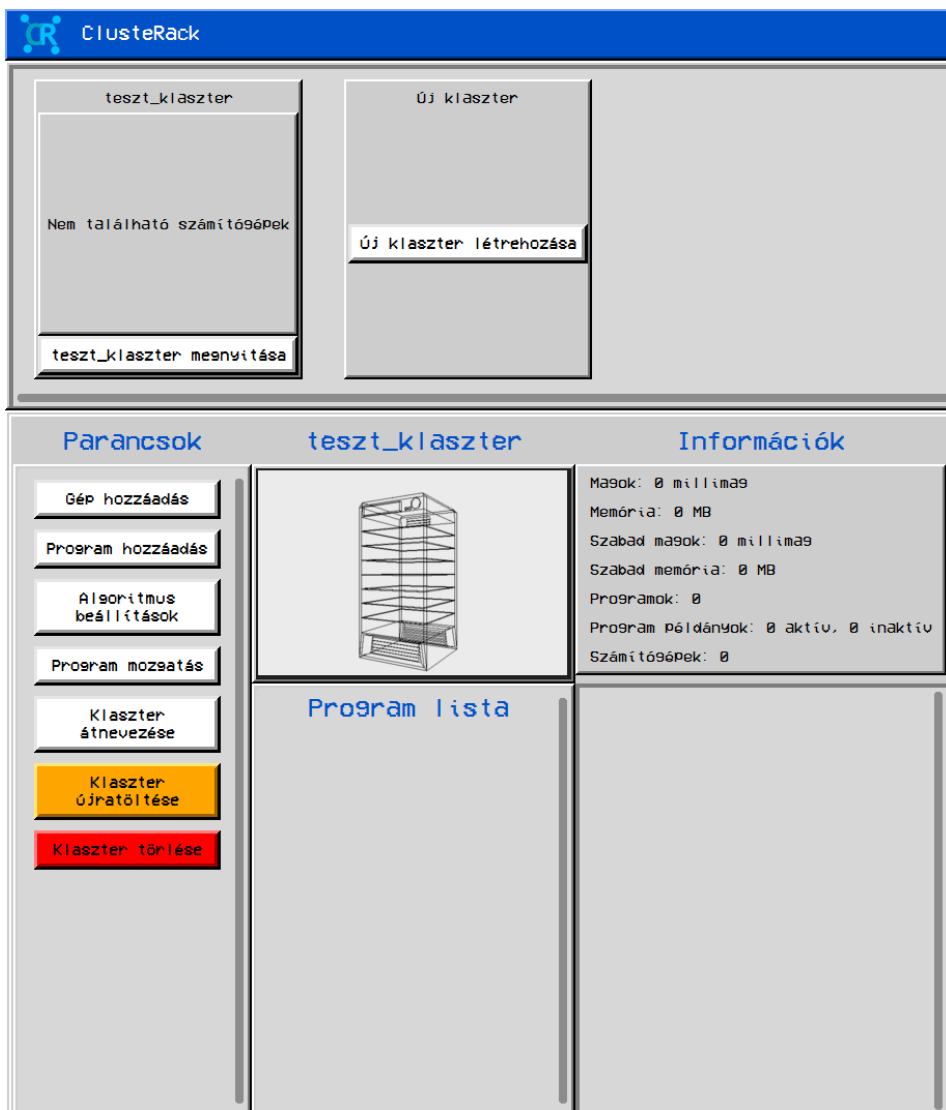
Az “Új klaszter létrehozása” gombra kattintva megjelenik egy felugró ablak. Itt meg kell adni a klaszter nevét.



A “létrehozás” gomb megnyomása után létrejön a klaszter és megnyílik a klaszter nézet.

Klaszter Nézet

A klaszter nézet tartalmazza a kiválasztott klaszter funkcióit és információit.



Bal (Parancsok) oszlop:

Itt találhatóak a gombok a klaszter kezeléséhez.

Középső oszlop:

Itt láthatunk egy 3D modellt a klaszterről. Annyi gépet jelenít meg, amennyit tartalmaz, de maximum 8 lehet benne.

Program lista: megjeleníti a .klaszter fájlban található programokat és itt tudjuk majd kezelni azokat.

Jobb (Információk) oszlop:

Megmutatja az információkat a klaszterről és alul a gépeket.

Gép hozzáadás:

Ezzel a gombbal lehet új számítógépet hozzáadni a klaszterhez. Meg kell adni a nevét és a rendelkezésre álló erőforrásait. A "létrehozás" gomb lenyomása után újratölti magát a felület és megjelenik a gép.

The screenshot shows a software interface titled "ClusteRack Subwindow". On the left is a vertical sidebar with several buttons: "Gép hozzáadás" (highlighted), "Program hozzáadás", "Algoritmus beállítások", "Program mozaik", "Klaszter átnevezése", "Klaszter újratöltése" (yellow), and "Klaszter törlése" (red). The main area of the window is titled "Gép hozzáadás a teszt_klaszter klaszterhez". It contains three input fields: "Számítógép neve:" with the text "test9ep", "Magok száma millimagokban:" with the value "1000", and "Memória megabájtban:" with the value "1000". A blue button labeled "Létrehozás" is positioned below the input fields.



teszt_klaszter

teszt_klaszter megnyitása

Új klaszter

Új klaszter létrehozása

Parancsok

teszt_klaszter

Információk

Gép hozzáadás

Program hozzáadás


Algoritmus beállítások

Program mozgatás

Klaszter átnevezése

Klaszter újratöltése

Klaszter törlése



Program lista

Magok: 1000 millimag
Memória: 1000 MB
Szabad magok: 1000 millimag
Szabad memória: 1000 MB
Programok: 0
Program példányok: 0 aktív, 0 inaktív
Számítógépek: 1

testgep

testgep megnyitása

Program hozzáadás

Amennyiben egy új programot akar hozzáadni a klaszterhez, azt itt tudja elvégezni. A felugró ablakban a következőket kell megadni: program neve, példányszám, szükséges magok száma millimagokban és szükséges memória megabájtban. A "Program hozzáadásakor" létrejön a program és bekerül az adott klaszter .klaszter fájlába.

Gép hozzáadás

Program hozzáadás

Algoritmus beállítások

Program mozsatás

Klaszter átnevezése

Klaszter újratöltése

Klaszter törlése

ClusteRack Subwindow X

Program hozzáadása a `teszt_klaszter` klaszterhez

Program neve

Futtatandó példányok:

Magok száma millima9okban:

Memória me9abájtban:

Program hozzáadása

Ezek után megjelenik egy másik ablak, hogy a program nem fut elég példányszámban, tehát észlelte a klaszter és létre kell hozni belőle példányokat.

Amennyiben az 1. opciót választja, a klaszter létrehozza és kiosztja a példányokat, ha a 2. opciót, akkor a klaszter nem hoz létre példányokat belőle, de elmenti a programot

Gép hozzáadás

Program hozzáadás

Algoritmus beállítások

Program mozsatás

Klaszter átnevezése

Klaszter újratöltése

Klaszter törlése

ClusteRack Subwindow X

Program hozzáadása a `teszt_klaszter` klaszterhez

Program neve

Futtatandó példányok:

Magok száma millima9okban:

Memória me9abájtban:


Program hozzáadása

Nem fut elég `'testProgram'` példány, minimum 2 kell!

Szeretne újakat indítani?

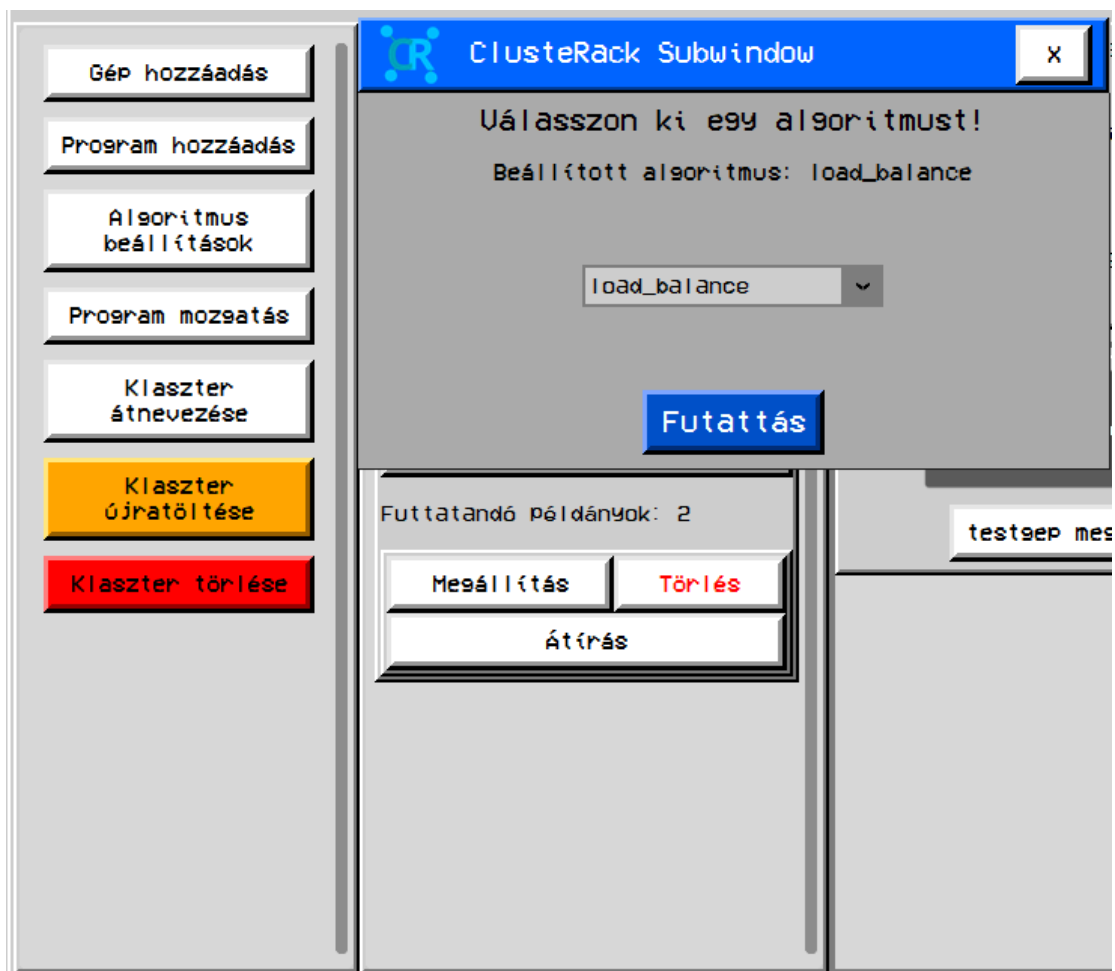
1 - Igen

2 - Mégse

Parancsok	teszt_klaszter	Információk
<div>Gép hozzáadás</div> <div>Program hozzáadás</div> <div>Algoritmus beállítások</div> <div>Program mozsatás</div> <div>Klaszter átnevezése</div> <div>Klaszter újratöltése</div> <div>Klaszter törlése</div>	<div>  </div> <div> <h3>Program lista</h3> <div>TESTPROGRAM</div> <div>Futtatandó példányok: 2</div> <div> <div>Megállítás</div> <div>Törlés</div> </div> <div>Átírás</div> </div>	<div> Magok: 1000 millima9 Memória: 1000 MB Szabad magok: 980 millima9 Szabad memória: 980 MB Programok: 1 Program példányok: 2 aktív, 0 inaktív Számítógépek: 1 </div> <div> <div>test9ep</div> <div>test9ep megnyitása</div> </div>

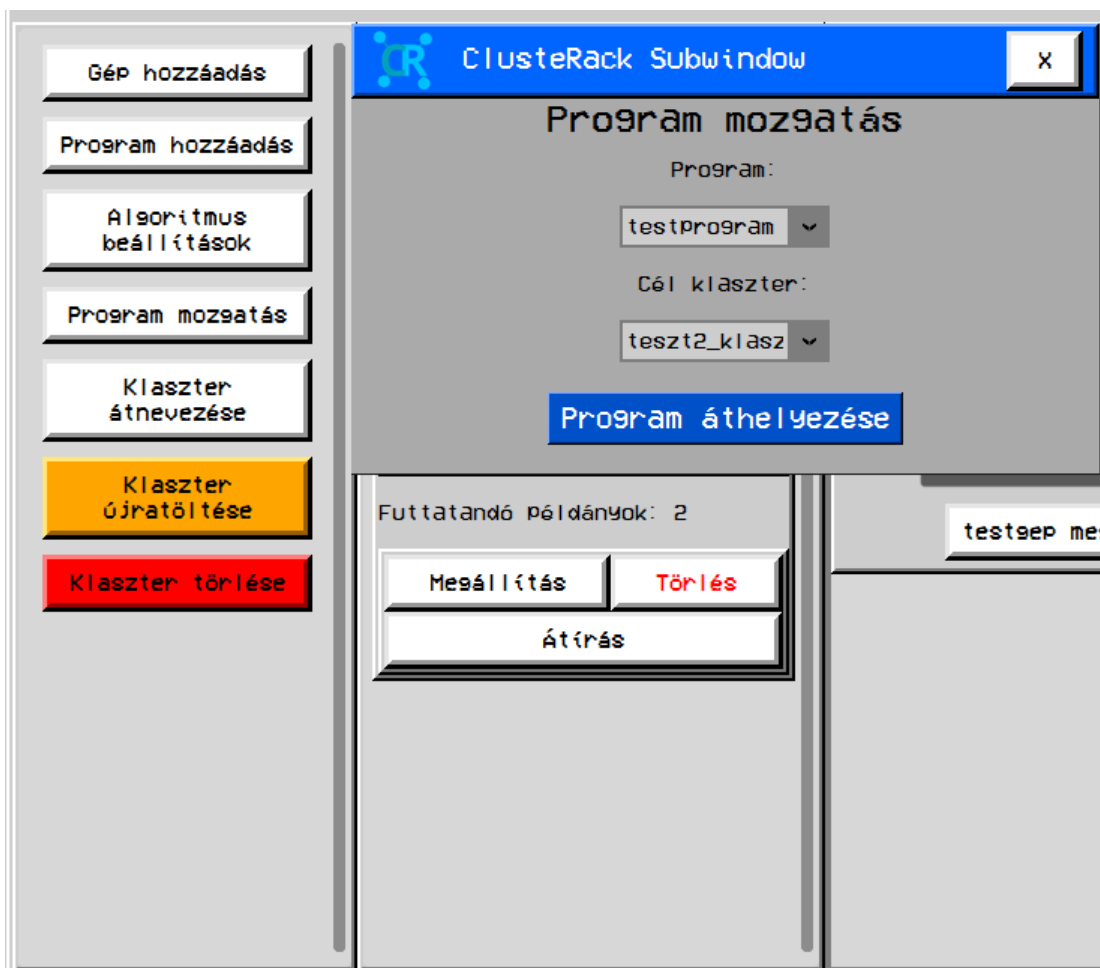
Algoritmus beállítások gomb:

A klasztereknek 3 fajta programpéldány elosztó algoritmusuk van. Ezek pontos működését a fejlesztői dokumentációban találja meg. Az algoritmusok a következők: load_balance, best_fit és fast. Ezek közül a legördülő menüben tud választani és a "futtatás" gombbal tudja lefuttatni. A felület frissül és utána meg tudja nézni, hogy változott-e a programpéldányok elrendezése a gépeken.



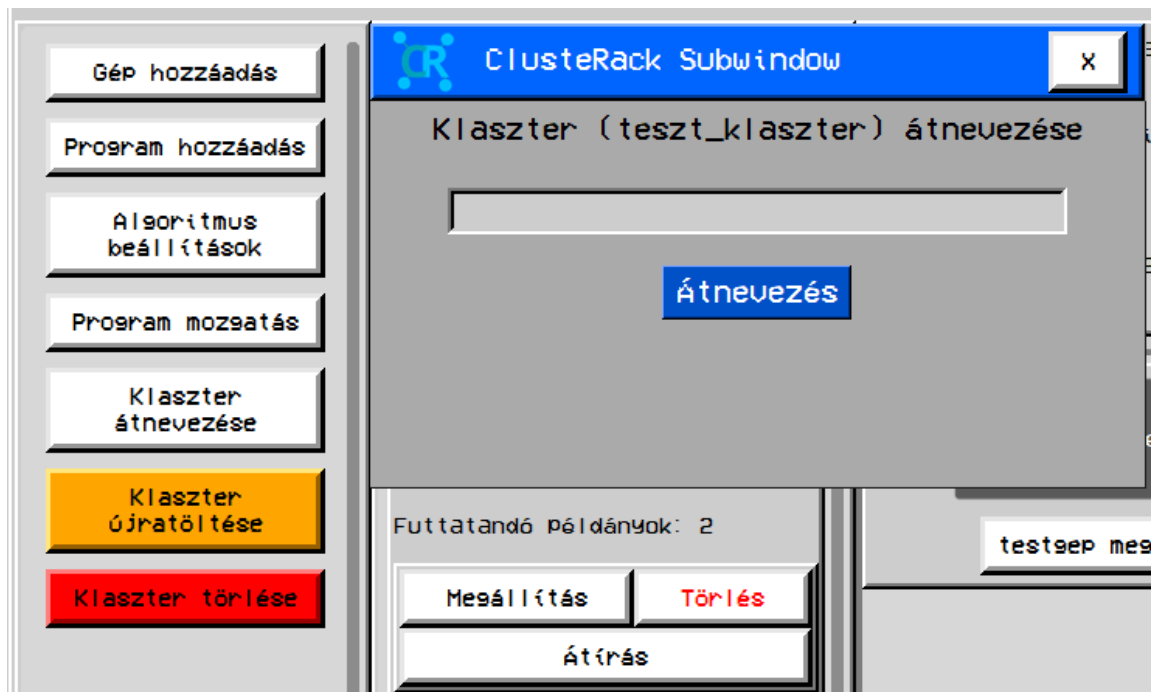
Program mozgatás:

Rendszerünk képes programokat mozgatni klaszterek között. Ezen gomb lenyomásakor megnyílik egy újabb felugró ablak, ahol kiválaszthatja a programot, amit át akar helyezni, és a cél klasztert.



Klaszter átnevezés:

A kiválasztott klaszter nevét is meg lehet változtatni. Itt csak az új klaszter nevet kell megadni.



Klaszter újratöltése: ez a gomb újratölti a klasztert és klaszter nézetet is.

Klaszter törlése: ezzel a gombbal törölheti a klasztert.



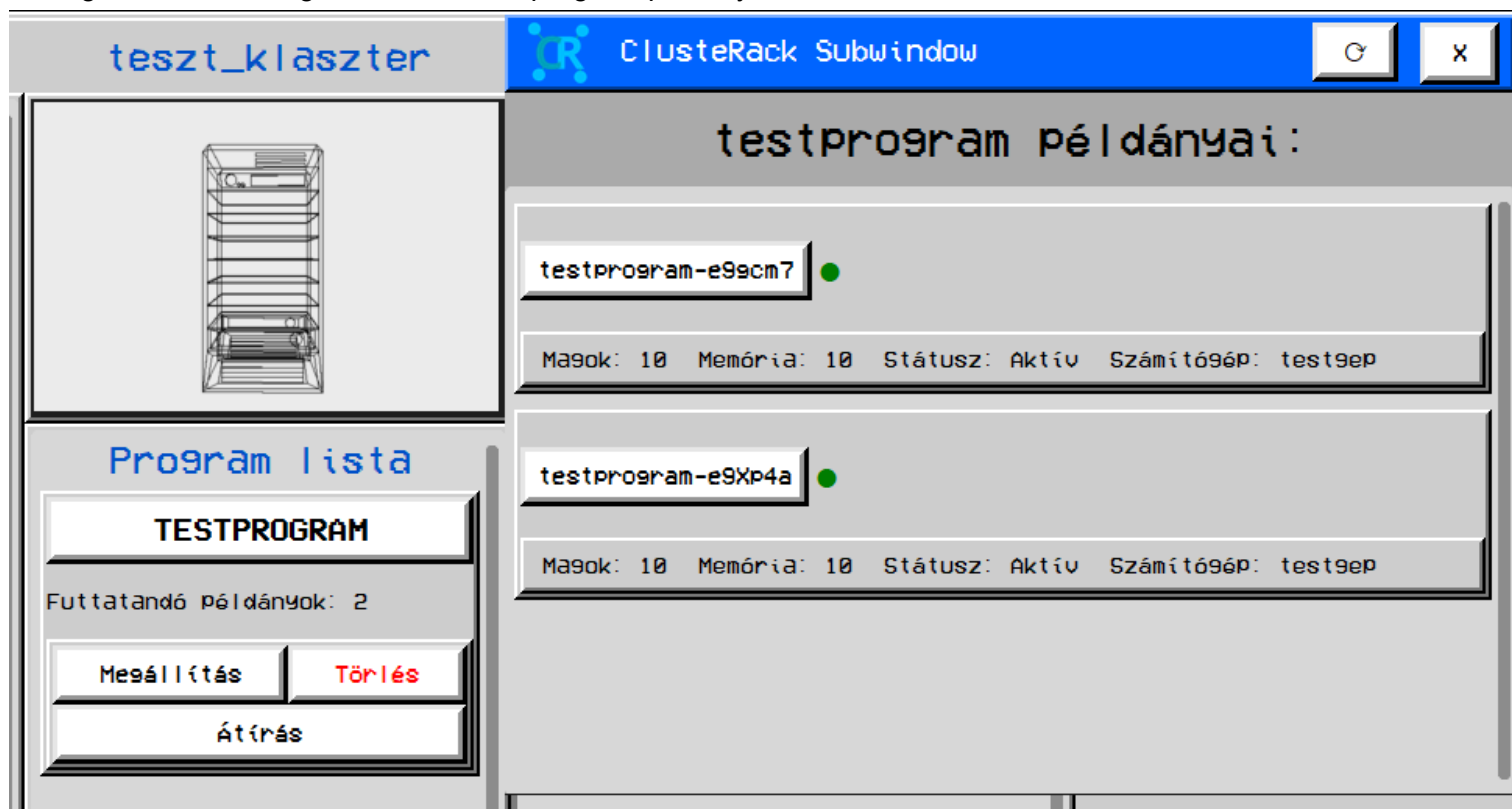
A programlista alatt találja meg a klaszteren lévő aktív és inaktív programokat.

Megállítás gomb: Átírja az adott programpéldány számát nullára, így állítva le a programot. Nem törli őket.

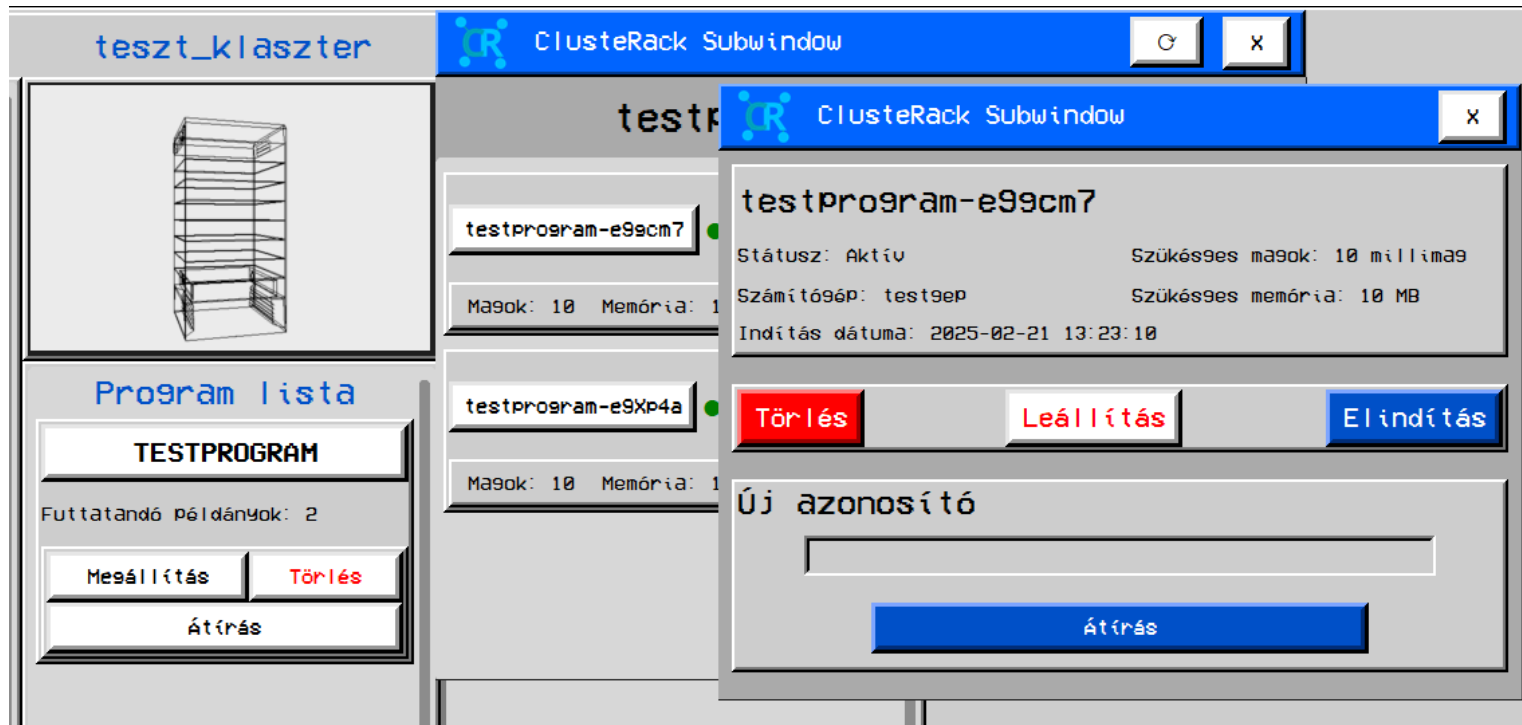
Törölés gomb: törli az adott programot és példányait a klaszterről és annak gépeiről.

A **program nevére** kattintva:

A felugró ablakban megnézheti az adott program példányait.

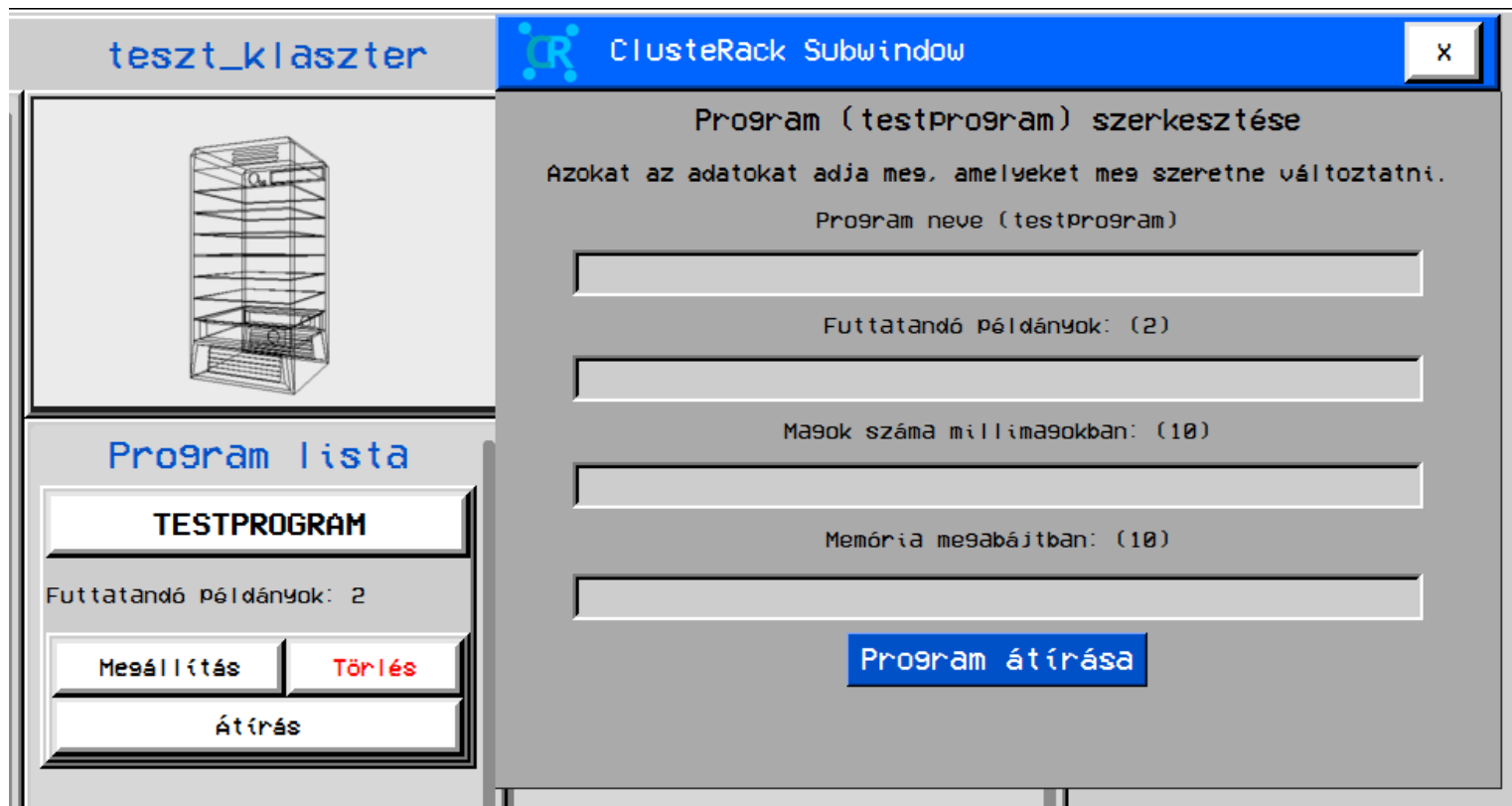


A nevükre kattintva megnyithatja egy külön ablakban az adott programpéldányt, ahol egyaránt tudja kezelni is.



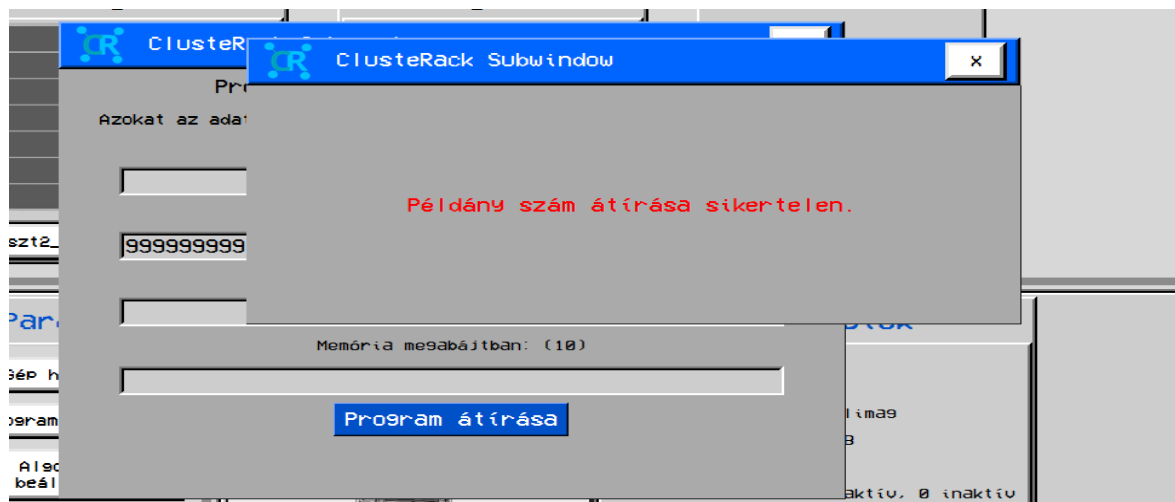
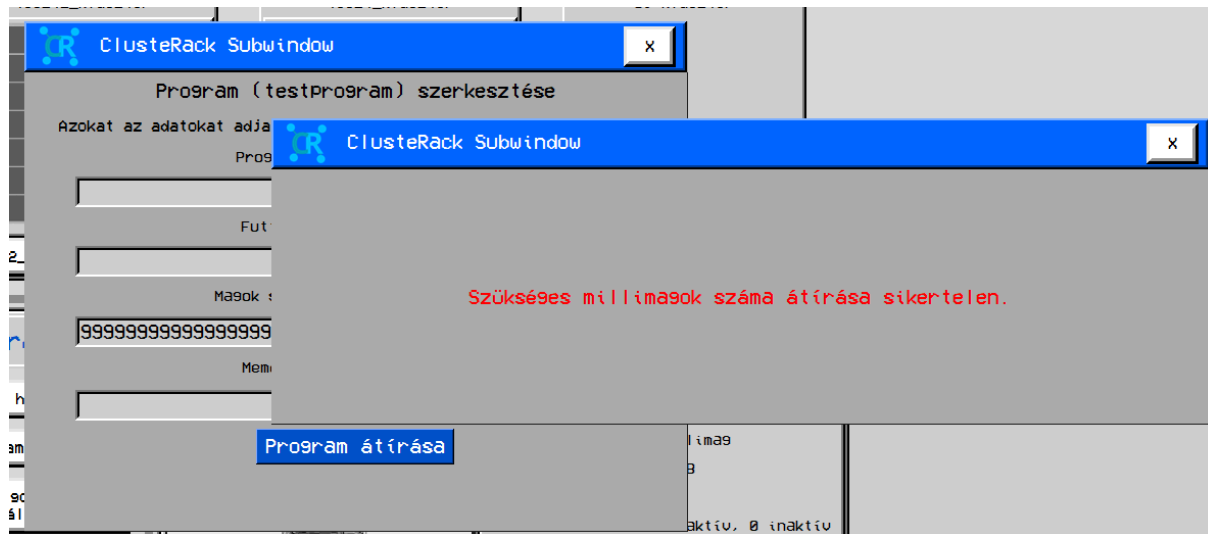
Átírás:

Ez a gomb ad lehetőséget a program adatai módosítására.

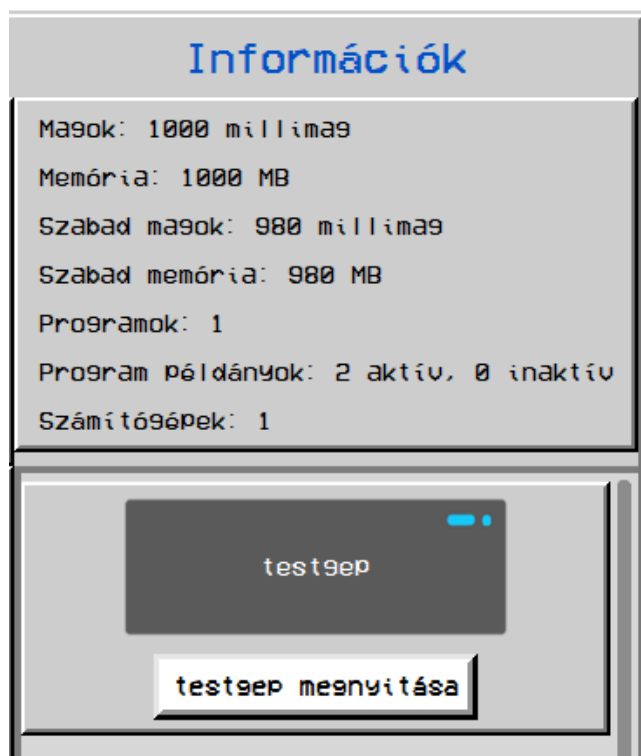


Itt ahogy a felugró ablakon is olvasható, csak annak a tulajdonságnak kell értéket megadni, amit meg akar változtatni. A “program átírása” gomb megnyomása után a klaszter elmenti a változtatásokat.

KIEGÉSZÍTÉS: felléphetnek hibák, hogyha túl nagy példányszámot vagy túl nagy futtatáshoz szükséges memóriát és/vagy magszámot ad meg.



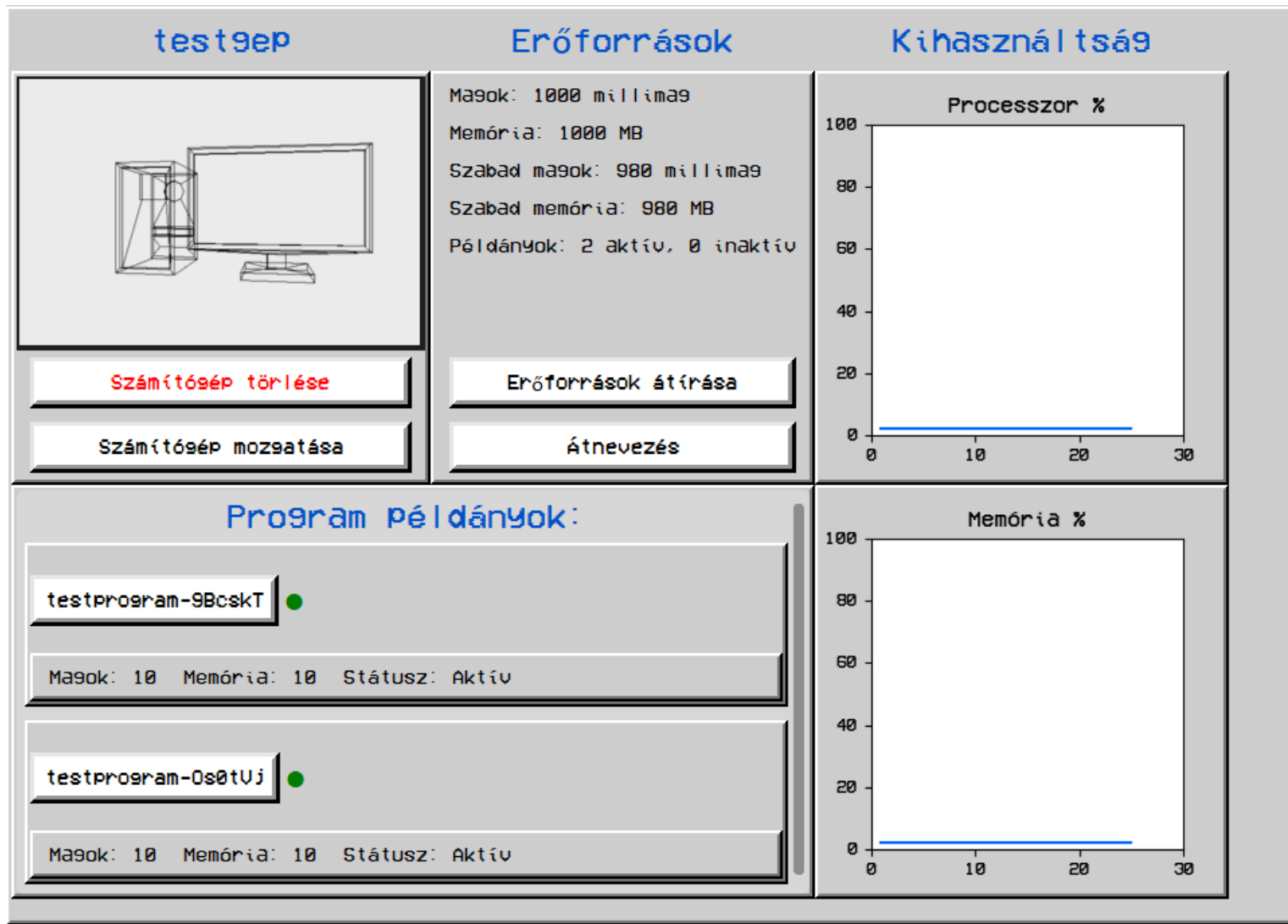
Jobb (Információk) oszlop



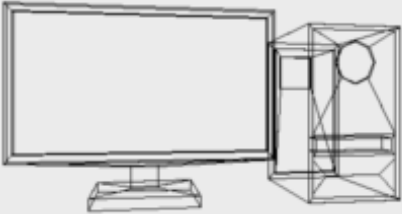
Itt láthatja a összegezve a rendelkezésre álló erőforrásokat, szabad erőforrásokat, programok és számítógépek számát és a programpéldányok aktív és inaktív számukat.

Alul láthatja a klaszter számítógépeit és itt tudja megnyitni a kiválasztott gépet számítógép nézetben, ahol kezelni tudja azt.

Számítógépnézet



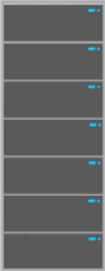
Ez a nézet megmutatja egy adott gép erőforrásait, rajta futó programpéldányokat és a kihasználtságát.

testgép	Erőforrások
	Magok: 1000 millima9 Memória: 1000 MB Szabad magok: 980 millima9 Szabad memória: 980 MB Példányok: 2 aktív, 0 inaktív
Számítógép törlése	Erőforrások átírása
Számítógép mozgatása	Átnevezés

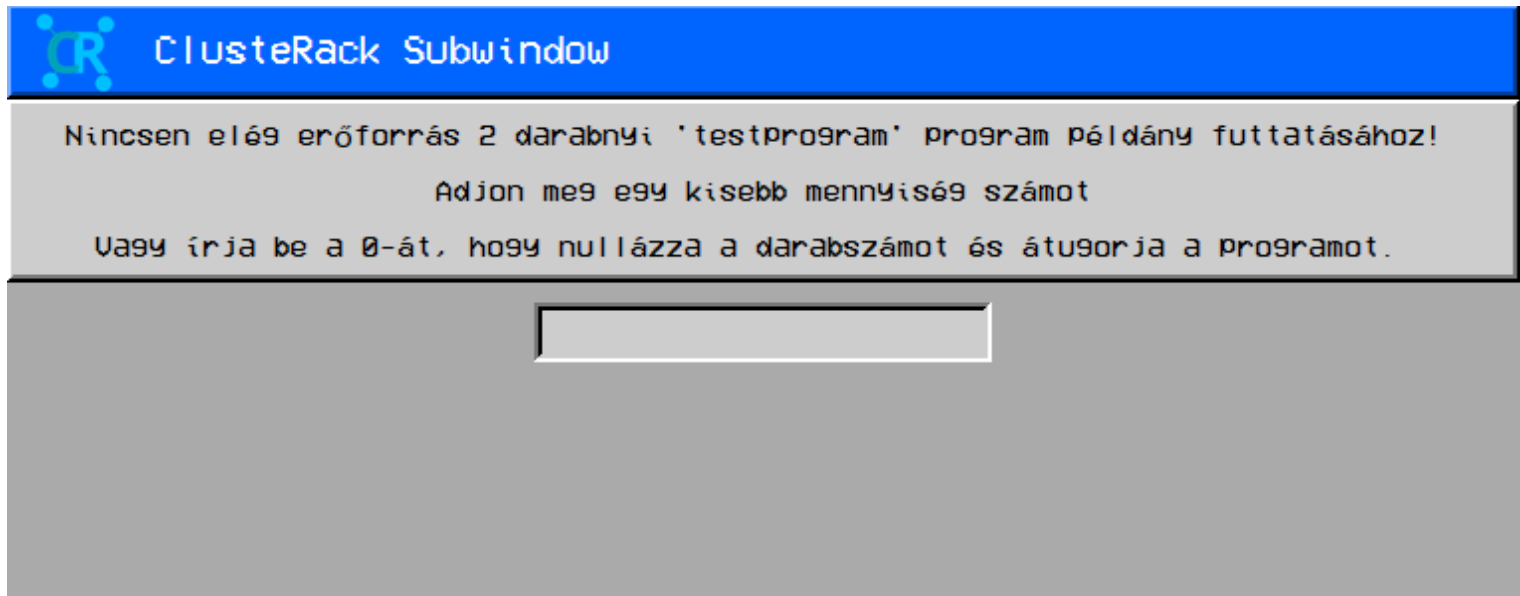
Számítógép törlése: törli a kiválasztott gépet a klaszterből

Számítógép mozgatása:

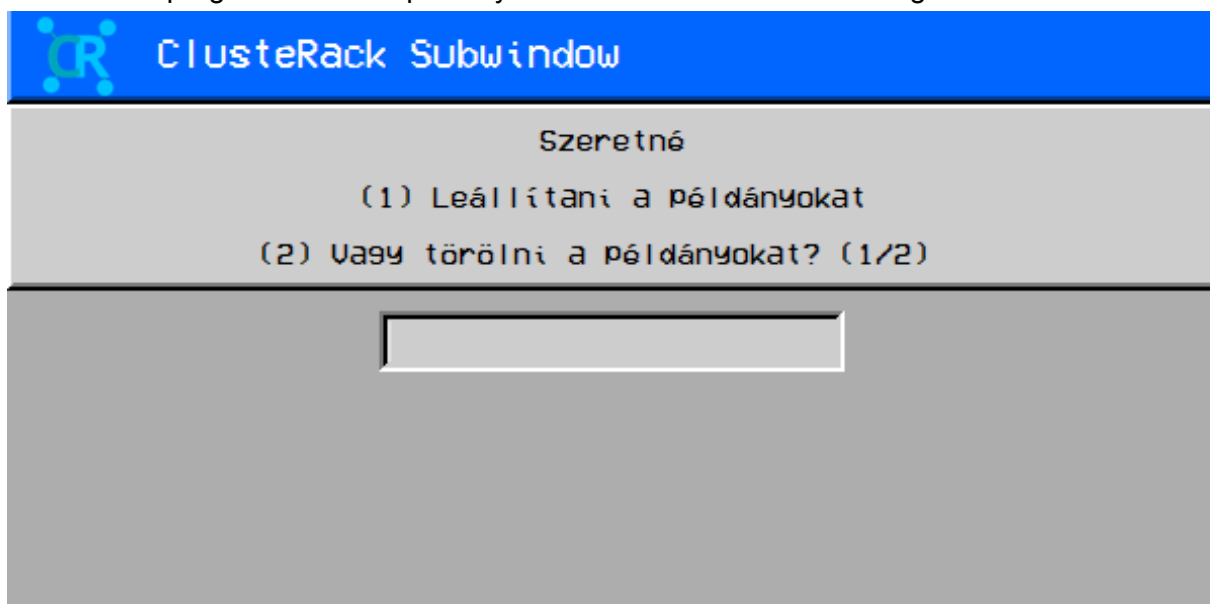
A rendszer számítógépeket is tud mozgatni, de a programpéldányokat előtte törli róla, mert a klaszterek klaszter fájjai nem biztos, hogy megegyeznek. Törlés után a klaszter és a cél klaszter újraosztja a példányokat.

teszt_klaszter	teszt_klaszter2	ClusteRack Subwindow
	Nem található számítógépek	testgép áthelyezése Cél klaszter: teszt_klaszter teszt_klaszter2 Számítógép áthelyezése
teszt_klaszter megnyitása	teszt_klaszter2 megnyitása	

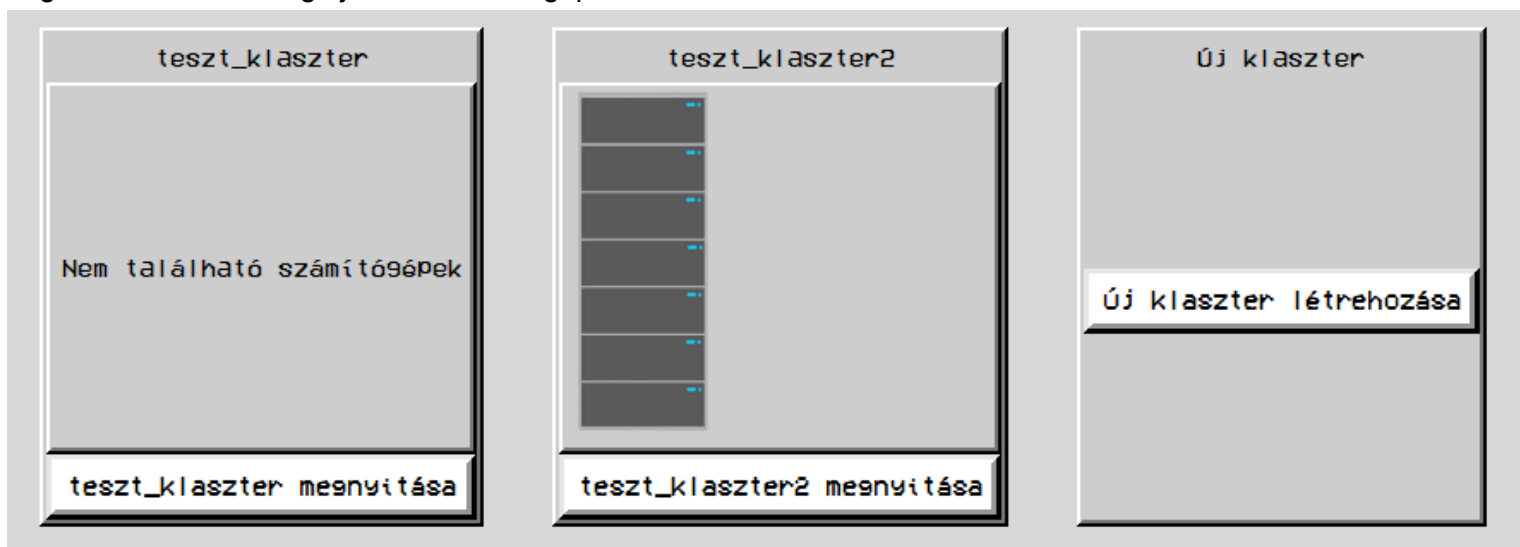
FONTOS: Ha a klaszternek nem marad elég erőforrása, akkor szól:



Ezek után a program leveszi a példányszámot és a következő ablak ugorhat fel:



Végül sikeresen átmozgatja a rendszer a gépet.

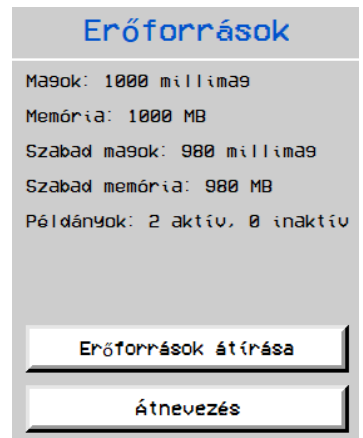
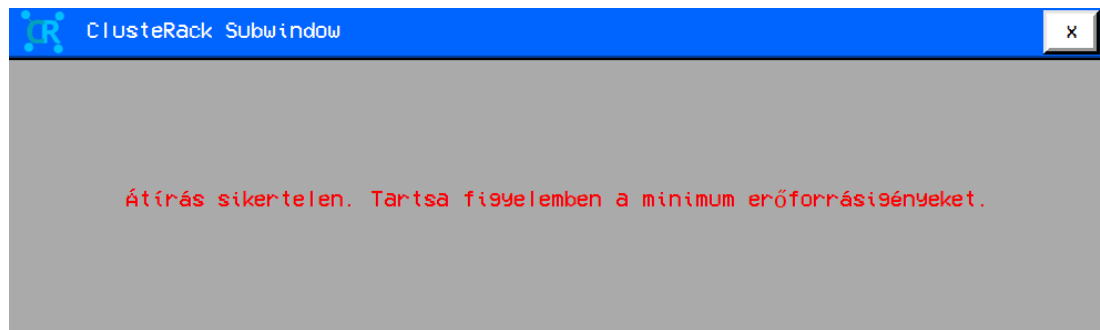
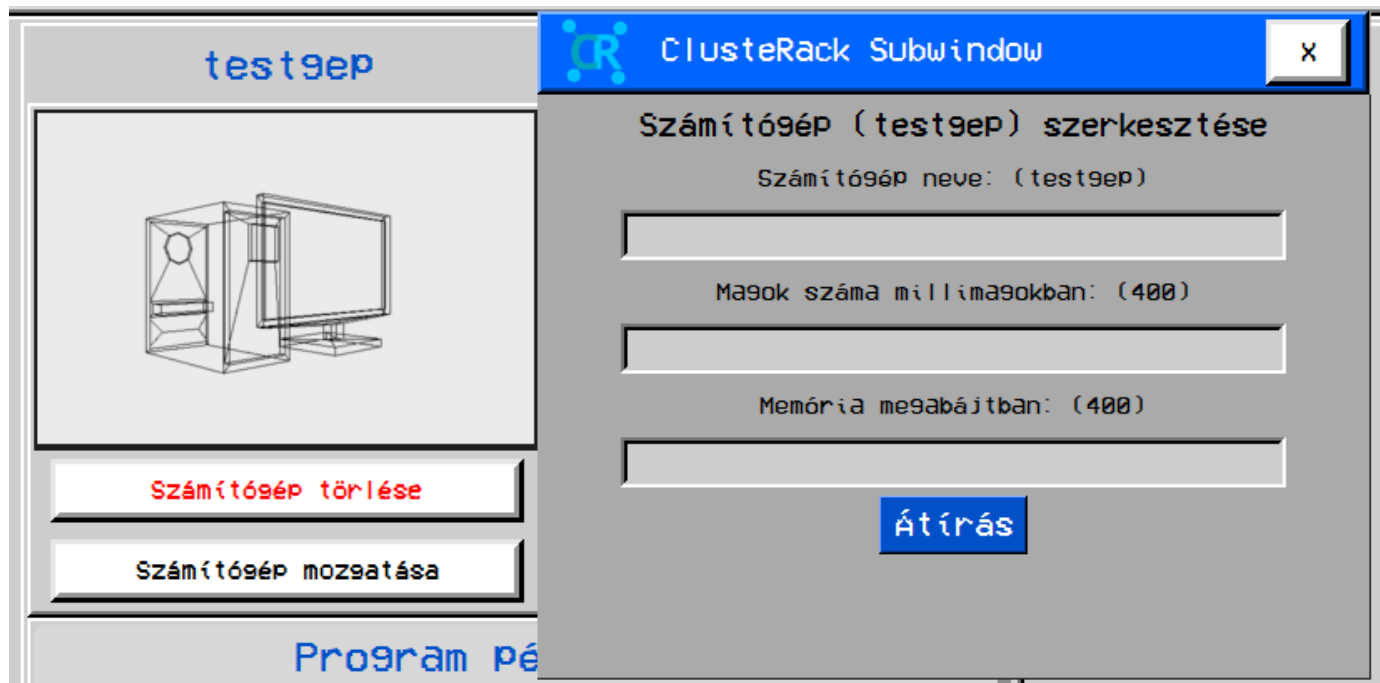


Erőforrások átírása:

A program biztosítja azt is, hogy bármikor meg tudja változtatni a számítógépek erőforrásait.

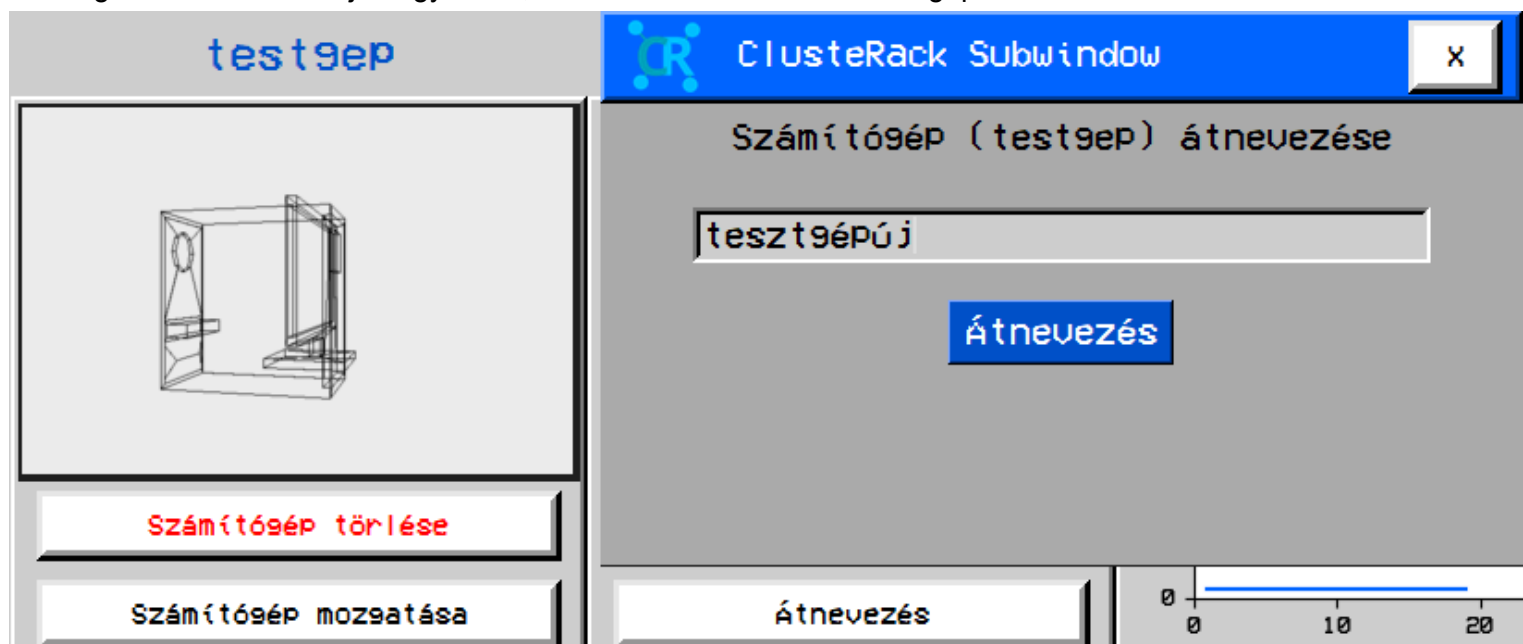
Az egyszerűség kedvéért ki van írva a mostani erőforrás értékei is.

A rendszer odafigyel, hogy ne lehessen túlterhelni a gépet.



Átnevezés:

Erre a gombra kattintva feljön egy ablak, ahol át lehet nevezni az adott gépet.



tesztgép	Erőforrások
	Magok: 1000 millima9 Memória: 1000 MB Szabad magok: 980 millima9 Szabad memória: 980 MB Példányok: 2 aktív, 0 inaktív
<div>Számítógép törlése</div>	<div>Erőforrások átírása</div>
<div>Számítógép mozgatása</div>	<div>Átnevezés</div>

Példányok kezelése

A programpéldány lista tartalmazza a számítógépre kiosztott példányokat.

Program példányok:


teszt-HyKbI ●

Magok: 10 Memória: 10 Státusz: Aktív

teszt-MPvIS2 ●

Magok: 10 Memória: 10 Státusz: Aktív

Itt ki is lehet választani a példányokat, ha rákattint a nevükre..

 ClusteRack Subwindow

teszt-HytKbI

Státusz: Aktív Szükséges ma9ok: 10 millima9
 Számítógép: tesztgépűj Szükséges memória: 10 MB
 Indítás dátuma: 2025-02-21 18:47:02

Törles Leállítás Elindítás

Új azonosító

 Átírás

Törles: törli az adott program példányát.


Leállítás: leállítja az adott programpéldányt. Nem törli.

Elindítás: Ha le van állítva a példány, elindítja azt.

Azonosító:

Lehetőség van új azonosító generálására. Ezt lehet:

- Automatikusan végrehajtani: ha üresen hagyja a mezőt és egyből a gombra kattint
- Manuálisan átírni: ha kitölti a mezőt és utána kattint a gombra. Ilyenkor több dologra is oda kell figyelni: egyéni azonosító legyen és feleljen meg az alap feladatban leírtaknak.(6 karakter hosszú és alfanumerikus)

 ClusteRack Subwindow

teszt-UkNWPF

Státusz: Aktív Szükséges ma9ok: 10 millima9
 Számítógép: tesztgépűj Szükséges memória: 10 MB
 Indítás dátuma: 2025-02-21 18:47:02

Törles Leállítás Elindítás

Új azonosító

 Átírás

 ClusteRack Subwindow

teszt-ididid

Státusz: Aktív Szükséges ma9ok: 10 millima9
 Számítógép: tesztgépűj Szükséges memória: 10 MB
 Indítás dátuma: 2025-02-21 18:47:02

Törles Leállítás Elindítás

Új azonosító

 Átírás

A Konzolon keresztül:

Amennyiben a konzolon keresztül szeretne dolgozni, a dokumentációban alul több információ van megemlítve a megnyitásról, használatról és a parancsokról.

Használt technológiák:

- **Git**

Verzió követés szempontjából azért esett a git-re a választásunk mivel minden csapattagunk járatos a használatában. A verziókövetés sokat segített nekünk a projekt készítése során.

- **VSCode**

Egyszerű használata és múltbéli tapasztalataink miatt ezt a programozási környezetet választottuk.

- **Python 3.13.2**

Mivel minden könyvtárunk támogatta a python ezen verzióját és ez egyben a legújabb is, ezért ezt láttuk a legjobbnak.

- **Krita**

Egy viszonylag egyszerűen kezelhető rajzprogram, amivel hamar el tudtuk készíteni a szükséges képeket.

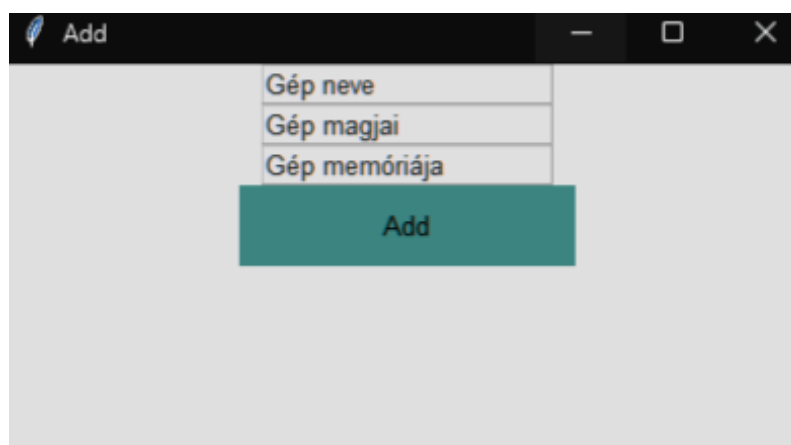
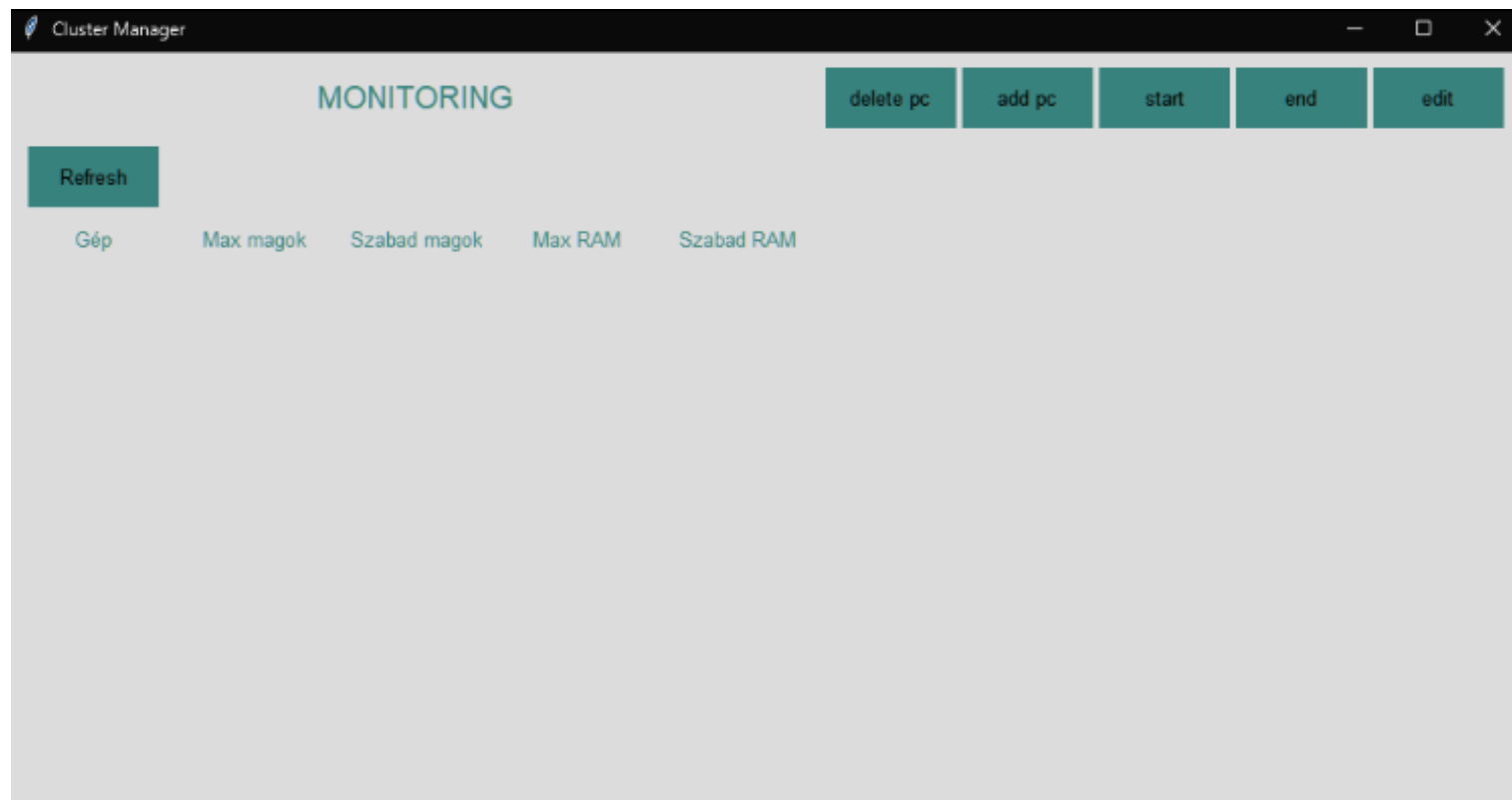
- **Blender**

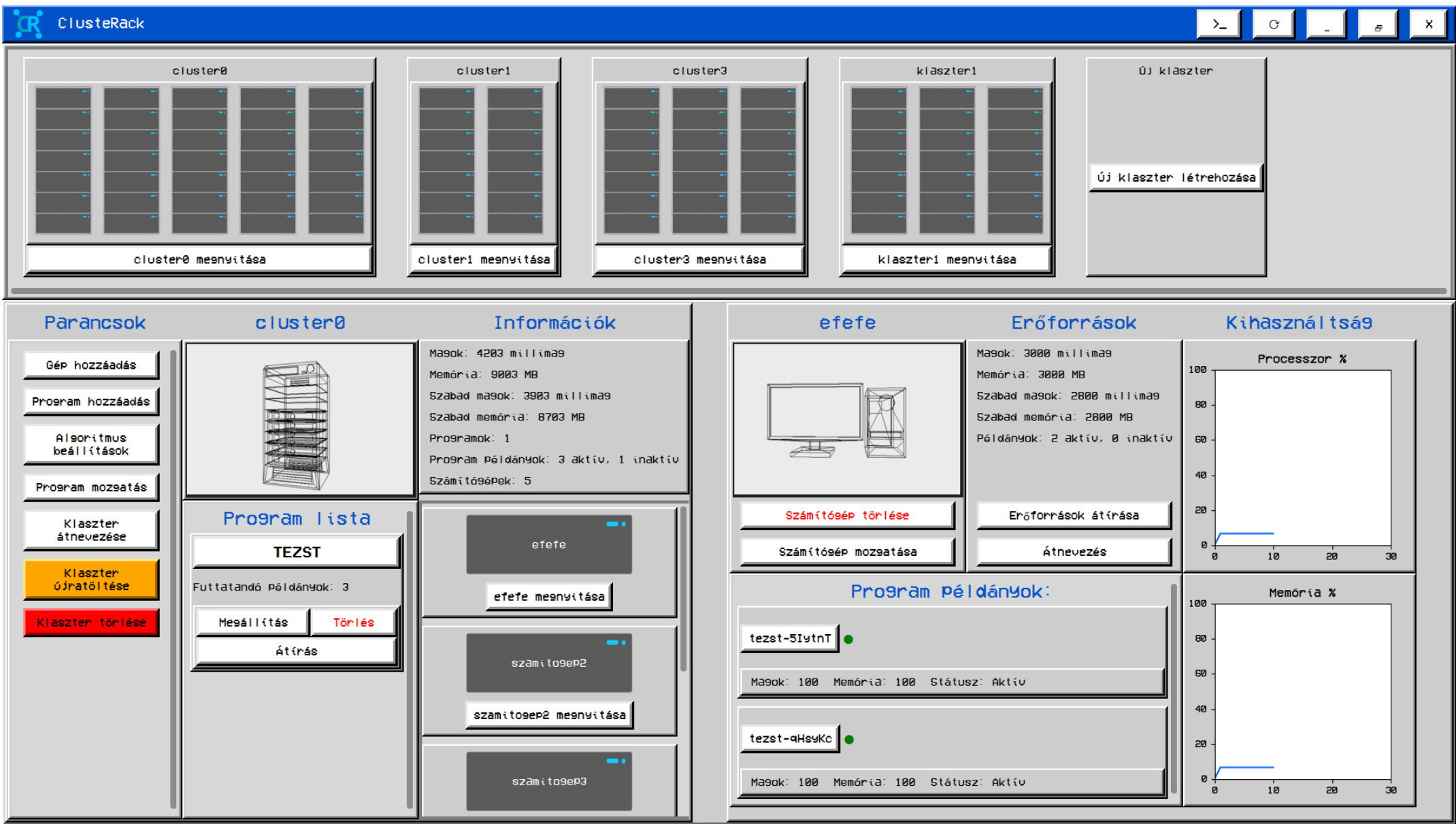
Egy nyílt forráskódú 3D modellező csomag. Az felhasználói felületen a render motorunkon belül megjelenített modellek elkészítéséhez használtuk. Azért választottuk mivel a csapatnak volt már tapasztalata a használatában.

A program újraírása

Az első fordulót követően a programot többször is átírtuk, hogy stabilabb és könnyebben használható legyen

Képek az első verzióból





 ClusteRack Subwindow X


tezst-5IytnT

Státusz: Aktív Szükséges magok: 100 millima9
Számítógép: efefe Szükséges memória: 100 MB
Indítás dátuma: 2025-02-20 19:32:56

Törlés Leállítás Elindítás

Új azonosító

Átírás

 ClusteRack Subwindow X

Gép hozzáadás a cluster0 klaszterhez

Számítógép neve:

Magok száma millimagokban:

Memória megabájtban:

Létrehozás

Első dolgunk volt egy nevet adni a programnak, ami a ClusteRack lett. Az újraírást követően jelentős mértékben meggyorsult a fejlesztés és a Windows 95-ös kinézetet vettük ihletül.

A régi verzióhoz képest nagyjából 26 új funkcióval bővült ki csak a backend.

Felhasznált szoftver eszközök

Könyvtárak:

[Tkinter](#)
[Customtkinter](#)
[Matplotlib](#)
[Pygame](#)
[Colorama](#)
[Software 3D engine](#)
[Tkinter Menu Bar](#)

A kezelőfelületen látott képeket Kovács Dániel, a 3D modelleket Tóth Péter készítette.


A "Software 3D engine"-t át kellett írunk, hogy kompatibilis legyen a többi könyvtárral. A license-t megtalálja a "licenses.txt" fájlban.

Funkciók technikai működése

Gyökér szintű hozzáférés

A program egy mappa alatt tartja a klasztereket ez a gyökérmappa. A gyökérmappa képes klaszterek között számítógépeket és programokat mozgatni. Segítségével egyszerre több klaszterhez is hozzáférhetünk és változtathatjuk azokat.

Egyéni parancsértelmező (Custom Command Interpreter)

A CCI a GUI felső részén található, ha rákattintunk a CCI ikonjára (), megnyílik egy terminál, ahol meg lehet adni különböző parancsokat. A parancs megadása után a program lefuttatja a parancshoz tartozó függvényt.

Különleges gombok és paraméterek:

- **Tab:** A "Tab" lenyomásával lehet használni az autocompletet, abban az esetben, hogy több javasolt parancs is lehetséges, kiírja a program a lehetséges parancsokat, amik közül a fel/le nyíl használatával lehet választani (enter lenyomásáig lehetséges)

```
[ROOT]: Starting cleanup...
[ROOT]: Cleanup completed. Removed a total of 0 incorrect files plus folders.
[ROOT]: Root (Thing) initialized succesfully with 5 cluster(s).

User>s
select
save_amount
save_all
User>s_
```

- **Nyilak:**
 - Bal/Jobb nyíl: A cursor balra/jobbra mozgatása
 - Fel/Le: Előzmények és javaslatok közül választás
 -
- **Delete:** A cursor utáni karakter törlése
- **?:** Ha magában van akkor leírja mit lehet abba a pozícióba írni

```
User>select cluster ?
apple apple
cluster0
cluster1
cluster2
cluster3
User>select cluster
```

- **Enter:** A parancs beküldése, ha,
 - a parancs nem egy teljes parancs, ahol lehetséges kiegészíteni ott kiegészíti magának:

```
User>se ro
selected the root
Thing>_
```

- A parancs nem egy teljes parancs, és több dologgal is ki lehet egészíteni, akkor ad javaslatokat:

```
Thing>se c cl t
cluster
computer
Thing>se c cl t_
```

- **Backspace:** A cursor előtti karakter törlése
- **/?:** Kiírja a parancs BNF-leírását

```
User>select /?
USAGE:
select <root>
select <cluster> <cluster_name>
select <computer> <cluster_name> <computer_name>

Options:
root Switches to root mode in the CLI.
cluster <cluster_name> → Switches to cluster mode for the specified cluster.
computer <cluster_name> <computer_name> → Switches to computer mode for the specified computer.
Press enter to continue
```

- **Home:** Visszarakja a kurzort a 0 pozícióra
- **End:** Az input végére rakja a cursort
- **Space:** Lehet használni megadott neveknek, sőt még lehet autocompletelni és javaslatként is kiválasztani, de beküldés előtt a név ne legyen idézőjelben, és csak az első szó legyen megadva, szóval:

```
User>select co ?
"apple apple"
"cluster 2"
cluster0
cluster1
cluster2
cluster3
User>select co_
```

```
cluster3
User>select co app → User>select co "apple apple"
```

Terheléelosztási algoritmusok

Három algoritmussal rendelkezik a program.

1. Teherelosztó

- Rangsorolja a gépeket magjaik és ramjaik szerint majd ezt hasonlóan megteszi a programpéldányokkal is. Ezek után kiszámol egy heurisztikus értéket minden számítógépnek ami megmutatja hogy mennyire van kihasználva az adott gép. Ezt követően a legkisebb értékkel rendelkező gép számára kiosztja a legnagyobb programpéldányt ami ráfér. Ezt a lépést addig ismételi amíg minden programpéldány ki nincs osztva.

Ez a leglassabb algoritmus viszont így minden számítógép közel azonos kihasználtsággal rendelkezik kiosztás után.

2. Szorosan pakoló (Greedy best fit decreasing)

- Hasonlóan rangsorolja a gépeket és a programpéldányokat mint a teherelosztó algoritmus, de ez megkeresi azt a számítógépet melyen a legkevesebb hely marad hátra a kiosztás után és hozzá párosítja a program példányát.

Ez az algoritmus gyorsabb mint a teherelosztó, de előfordulhat hogy kiosztás után nem minden gép lesz azonosan kihasználva. Akkor működik legjobban, ha a számítógépek megközelítőleg egyenlő manggal és memóriával rendelkeznek

3. Gyors (First fit)

- Ez az algoritmus is rangsorolja először a gépeket, majd az első elérhető programpéldányt kiosztja az első számítógépnek amire ráfér a példány.

Ez a leggyorsabb algoritmus a három közül de hátránya, hogy nem megfelelő hely használathoz vezet.

Új funkciók listája

Gyökér feature-ök:

Gyökér újratöltése

Klaszter törlése kétféle módon

Klaszter készítés

Klaszter átnevezés

Programok mozgatása klaszterek között

Gépek mozgatása klaszterek között

A gyökérmappa kitisztítása

Klaszter

Klaszter újratöltése

Teher elosztási algoritmus kiválasztása

Teher elosztási algoritmus futtatása

Gépek törlése kétféle módon

Gépek készítése

Gépek átnevezése

Számítógépek specifikációinak beállítása

Program hozzáadása

Program törlése

Program megállítása

Program specifikációk módosítása

Program átnevezése

Programpéldány hozzáadása

Programpéldány állapot módosítása

Programpéldány törlése

*Programpéldány ID-ének megváltoztatása
(manuálisan vagy auto generált)*

Gépek

Fájlok tisztítása

Osztályok

A programunk modulokból épül fel amik kapcsolatban vannak egymással ennek az alapja a Root osztály

Root

Tartalmazza a valid klasztereket. A következő funkciókra képes:

- Gyökér újratöltése - Újra inicializálja a teljes programot elkezd a tisztítást.
- Klaszter törlése kétféle módon - Törölhet klasztereket gyenge módon (azaz ha a klaszter tartalmaz gépeket nem törlődik) és erős módon (a klaszter mindenképpen törlődik a nem figyelve az alatta) lévő
- Klaszter készítés - Új klasztereket tud létrehozni a gyökérmappában
- Klaszter átnevezés - A meglévő klasztereket át lehet nevezni
- Programok mozgatása klaszterek között - Az eredeti helyről kitöröljük majd ha még nem létezik ilyen program a cél klaszteren elindítjuk a programot rajta
- Gépek mozgatása klaszterek között - Hasonlóan a programok mozgatásához ha a mozgatni kívánt gép nem létezik a cél klaszteren átvisszük.
- A gyökérmappa tisztítása - Ellenőrzi a gyökér mappát, átnézi, megkeresi az ismeretlen fájlokat majd a felhasználó választása alapján törli vagy ignorálja azokat

Cluster

Tartalmazza a valid gépeket és programokat.

A következő funkciókra képes:

Klaszter újratöltése - Újra inicializálja az adott klasztert, elkezd a klaszter tisztítását.

Tehereosztási algoritmus kiválasztása - Lehet választani a három tehereosztási algoritmus közül

Teher elosztási algoritmus futtatása - Szétosztja a klaszteren lévő számítógépeken a programokat a kiválasztott algoritmu salapján.

Gépek törlése kétféle módon - A kétféle törlési mód a gyenge (ha a tartalmaz éppen programpéldányokat), erős (a törlés nem veszi figyelembe gépen tartott fájlokat mappákat). Ha egy gépet úgy törölünk, hogy programpéldányok voltak rajta, akkor ezek a példányok áthelyezésre kerülnek a klaszterben. (Újra kiosztásnál az adataik ugyanazok maradnak).

Gépek készítése - Gépeket tudunk létrehozni új specifikációkkal.

Gépek átnevezése - A klaszteren lévő gépeket át lehet nevezni azok újraindítása nélkül.

Számítógépek specifikációinak beállítása - Egy számítógépen meg lehet változtatni az elérhető magok számát és a memória nagyságát.

A programok

Program hozzáadása - A klaszteren lehet új programokat indítani tetszőleges specifikációkkal

Program törlése - A klaszterről le lehet törölni programokat

Program megállítása - Egy programot meg lehet állítani. Ez a művelet inaktívvá teszi a program minden példányát

Program specifikációk módosítása - A program mag és memória használatát valamint a példányok számát lehet állítani

Program átnevezése - Egy program a klaszteren átnevezhető

Programpéldány hozzáadása - Egy programhoz lehet példányokat hozzáadni

Programpéldány állapot módosítása - A létező példányok státuszát meg lehet változtatni

Programpéldány törlése - Létező programpéldányok törölhetőek

Programpéldány ID-ének megváltoztatása (manuálisan vagy auto generált) - Egy létező példány azonosítóját meg lehet változtatni manuálisan vagy meg lehet engedni a programnak hogy automatikusan generáljon egyet.

A CMD parancsok BNF leírása

```
<command> ::= <user_command>
            | <root_command>
            | <cluster_command>
            | <computer_command>

<user_command> ::= "select" <root>
                  | "select" <cluster> <cluster_name>
                  | "select" <computer> <cluster_name> <computer_name>
                  | "select" <run_folder> <folder_name>
                  | "run" <file_name>
                  | "update_commands"
                  | "reload"
                  | "save_all" <file_name>
                  | "save_amount" <file_name> <amount>
                  | "exit"

<root_command> ::= "create_cluster" <cluster_name>
                  | "try_del_cluster" <cluster_name>
                  | "force_del_cluster" <cluster_name>
                  | "relocate_program" <program_name> <origin> <destination>
                  | "move_computer" <computer_name> <origin> <destination>
                  | "rename_cluster" <target_name> <new_name>
                  | "cleanup_root"
                  | "run" <file_name>
                  | "update_commands"
                  | "reload"
                  | "save_all" <file_name>
                  | "save_amount" <file_name> <amount>
                  | "exit"

<cluster_command> ::= "set_rebalance_algo" <algo_id>
                     | "run_rebalance"
                     | "create_computer" <computer_name> <cores> <memory>
                     | "try_del_computer" <computer_name>
                     | "force_del_computer" <computer_name>
                     | "rename_computer" <target_computer> <new_name>
                     | "edit_computer_resources" <computer_name> <cores> <memory>
                     | "get_cluster_programs"
                     | "get_cluster_instances"
```

```

| "start_program" <program_name> <instance_count> <req_cores> <req_memory>
| "kill_program" <program_name>
| "stop_program" <program_name>
| "edit_program_resources" <program_name> <prop_to_change> <new_value>
| "rename_program" <program_name> <new_name>
| "add_instance_gen_id" <program_name>
| "add_instance_user_id" <program_name> <instance_id>
| "edit_instance_status" <instance_id> <new_status>
| "kill_instance" <instance_id>
| "change_instance_id_gen" <instance_id>
| "change_instance_id_user" <instance_id> <new_instance_id>
| "cleanup_cluster"
| "run" <file_name>
| "update_commands"
| "reload"
| "save_all" <file_name>
| "save_amount" <file_name> <amount>
| "exit"

```

```

<computer_command> ::= "cleanup_computer"
| "run" <file_name>
| "update_commands"
| "reload"
| "save_all" <file_name>
| "save_amount" <file_name> <amount>
| "exit"

```

```

<root> ::= "root"

```

```

<cluster> ::= "cluster"

```

```

<computer> ::= "computer"

```

```

<folder_name> ::= <string>

```

```

<file_name> ::= <string>

```

```

<amount> ::= <integer>

```

```

<cluster_name> ::= <string>

```

```

<computer_name> ::= <string>

```

```

<target_name> ::= <string>

```

```

<new_name> ::= <string>

```

```

<program_name> ::= <string>

```

```

<instance_id> ::= <string>

```

```

<new_instance_id> ::= <string>

```

```
<prop_to_change> ::= <string>
<new_value> ::= <string>
<origin> ::= <string>
<destination> ::= <string>

<cores> ::= <integer>
<memory> ::= <integer>
<instance_count> ::= <integer>
<req_cores> ::= <integer>
<req_memory> ::= <integer>
<algo_id> ::= "load_balance" | "best_fit" | "fast"
<new_status> ::= <string>

<string> ::= ? any sequence of non-whitespace characters ?
<integer> ::= ? any sequence of digits ?
```