

Második programozás Házi Feladat

Skeleton

Adatok:

- név: Morvai Barnabás
- Neptun kód: EoHBoN
- feladat: Számítógép/perifériák
- leadás dátuma: 2024.05.05.

Egy új osztály

Létrehoztam egy új osztályt „User” néven, hogy a felhasználónak ne kelljen ismernie a szintaktikákat és a megvalósítás során történő változások okozta különbségeket. Így elég majd csak a tárolandó adattal foglalkoznia. Ez az osztály látja majd a processzort, a memóriát és a perifériákat is.

Skeleton

Az osztályok és a fontosabb (szinte az összes) függvények deklarációi létre lettek hozva. Emellett egy tesztprogram váza is elkészült a „gtest-lite.h” segítségével. Több különböző tesztet különítettem el, vizsgálva lesz a memória, a periféria és a hibakezelés funkcionalitása is. Ha minden kész, a korábbi jPorta feladatokhoz hasonlóan vége a tesztnek.

Megvalósítás

Nem csak deklaráció található a file-ok között. A generikus memória működése különösen érdekelt, szóval a Memória osztály nagy része késznek mondható. Bele lehet tenni, törölni lehet belőle adatokat, az adatok típusától függetlenül. Emellett ki lehet írni egy specifikus memóriacellát is, de ha az egészet szeretnék egyszerre látni, a memoryDump a memória összes elemét kilistázza. Ha érvénytelen adat van a cellában, akkor invalid jelzővel mutatja, hogy ott semmi nincs, minden más esetben a cella tartalma íródik ki szóközzel elválasztva.

A „gtest-lite.h” használatával csináltam egy külön tesztet az eddig megvalósított rész kipróbálására. Létrehoz különböző memóriacellákat, beleteszi a memóriába, kiveszi onnan, kiír egy-egy cellát majd végül az egész memória tartalmát.

Használat

A használhatóság érdekében létrehozott „User” osztály gondoskodik a felhasználóbarát kezelésről. Ha valaki perifériát csatlakoztat, egy egyszerű függvényhívással megteheti, ahova a kívánt adatot kell csak beírnia, onnantól a függvény vezérli le a periféria csatlakoztatását és az adat memóriába juttatását, ha a felhasználó azt szeretné.

Perifériák, HUB, Processzor

A HUB felelősége a perifériák feletti kontroll, hogy ne csak a levegőben lebegjenek szét a memóriában, hanem egy heterogén kollekcióban tartsa őket.

Másrészt, ha perifériából memóriába írás parancsot kap, ő az, aki kikeresi a megfelelő perifériapéldányt a heterogén kollekcióból és adja tovább annak adatát a processzor számára, aki majd eltárolja azt a memóriában.

A „User” osztály fogja meghívni a függvényeit, a felhasználó tudta nélkül, hogy neki ne kelljen tudnia, hogy működik a HUB és annak függvényei, szintaktikája.

A Processzor programozási szemmel egy felesleges elem, aki átveszi az adatot a perifériától és továbbítja azt a memória felé tárolás címszó alatt. Nyilván elhagyható lenne, de a valós modell megtartása érdekében van a lánc közepén. Esetleg, ha lenne szélessége a busznak, akkor a feldarabolását az adatnak ő végezné, mert akkor nem csak egy vector-nyi adatot kéne letárolásra küldeni, hanem annyit amennyi felé lett darabolva az az adatblokk.

Elvárt bemenet, kimenet

Bemenetként a felhasználó tudja megadni az adatokat, egyelőre kód formájában, de az egyszerű „User” osztályon függvényein keresztül.

Kimenetként a memória tartalmát tudja lekérni, akár egyesével, akár teljes memóriaképként.