

ORSI. 3. BEADANDÓ FELADAT

MIKUS MÁRK ISTVÁN

1. FELADAT

A bemenet input.txt első sorában egy N pozitív egész olvasható, ennyi diáknak kell biztosítani termet, míg a következő N sorban a hallgatók NEPTUN-kódjai, azaz N string követi egymást, ez mutatja a jelentkezési sorrendet.

Egy lehetséges bemeneti fájl:

```
7
OSAVH1
T0NDJB
4S1UPL
AXKAW4
22TQP7
NM8VPS
PJVNEU
```

Egy lehetséges kimeneti fájl:

```
22TQP7
4S1UPL
AXKAW4
NM8VPS
OSAVH1
PJVNEU
T0NDJB
```

1.1. Megjegyzések.

2. FELHASZNÁLÓI DOKUMENTÁCIÓ

2.1. **Környezet.** A program platformfüggetlen, a program futtatásához elég a futtatható állományok elindítása. Telepítésre nincs szükség.

2.2. Használat. A futtatáskor parancssori paramétert nem vár a program.

A futtatható állomány mellé kell helyezni az input.txt fájlt. Illetve ugyanide kell létrehozni egy üres output.txt nevű fájlt. A program az input fájlból fog dolgozni, majd az output fájlba írja az eredményt, ha tudja. Ha nem tudja manipulálni a megfelelő fájlokat, a program hibajelzést ad.

3. FEJLESZTŐI DOKUMENTÁCIÓ

3.1. A megoldás módja. A kódot logikailag két részre szeparáltuk, mester és gyermek folyamatra. A master.cpp forrásfájlban található főfüggvény (main) reprezentálja a mesterszálat, a programkonstrukciónkban. A gyermekfolyamatok implementálása a child.cpp-ben kap helyet. Itt kerül implementálásra, a megoldás magját adó rekurzív összefüggő rendezés-algoritmus. Illetve az ahhoz tartozó két függvény: a merge, amellyel sorozatokat rendezünk, és a mergeSort, amely az oszd meg és uralkodj elvén, szétosztja két újabb gyermek között az input sorozat felét.

A mester szálon lévő program kizárólagos feladata, az input fájlból az adatok beolvasása, majd azok továbbítása a gyerek felé. A végeredmény megkapásakor pedig, azt egy output fájlba kell írnia.

3.2. Implementáció. Az implementálás során a fentebb említett szemlélet volt a mérvadó. A program hibakezelést végez, a a fájlokkal való kommunikáció során(, majd később a pvm spwan metódusa kapcsán is). Az inputot a program egy string tömbbe pakolja be. Amely n elemű. A string tömböt egy szintén n-es forciklussal tölti fel.

Majd ugyanezen végigiterálva elküldjük a gyerekeknek a méretét, majd magát a tömböt is. A főfolyamat ezek után már csak visszavár egy n méretű string tömböt, amit miután sikeresen fogad, beleír az output fájlba.

A gyermekfolyamat törzse lényegében az üzenetek fogadásából áll, majd az eredmény előállításából és végül az eredmény visszaküldéséből.

Az eredmény előállítását a mergeSort nevű 3 paraméteres függvény végzi, amely rekuzívan hívható, s minden híváskor létrehoz 2 gyermeket, akiknek, a megfelelő szeletek méretét és magát a szeletet is átküldi, majd viszontvárja az eredményt.

3.3. Fordítás. A program forráskódja a már említett master.cpp illetve a child.cpp. A program fordításához követelmény egy c++11 szabványt támogató fordítóprogram megléte a rendszeren. A fejlesztés, ill. tesztelés során a g++ fordítót használtam. A fordítóban speciális beállításaként a stringeket támogató -Wno-write-strings kapcsolót is alkalmazni kell. Az std=c++11 kapcsoló is szükséges, mert alapértelmezetten régebbi c++ szabványt támogat a fordító. A programhoz csatolt Makefile.aimk fájl segítséget nyújt a fordításhoz, mert magába foglalja

a fordításhoz szükséges összes információt. Ennek segítségével elég meghívni az aimk, parancsot a forrássájlak könyvtárában állva. Majd miután végezett, futtatható is a program.

(PVM futtatása esetén: mostmár kiadható a pvm parancs, majd a pdm démonból indítható spawn-nal a master folyamat.)

3.4. Tesztelés. A tesztelést és a futtatást, PVM környezetben végeztem. Ehhez hozzáférhetőséget kaptam az ELTE Atlasz nnevű kiszolgálójához. A merge metódust, amely az összefésülést végzi, a kari honlapokon talált segédanyagok alapján implementáltam. Ahogy a mergeSort metódust a feladathoz kapcsolt, rekurzív leírás alapján került megvalósításra.

A merge metódushoz gyakorlati számításokat végeztem a szélsőséges tesztesetekre. (1,2 eleműek összefűzése) A mergeSort metódust, pedig futtatások során teszteltem, figyelve arra, hogy felépülésre kerül-e a megfelelő bejárési fa.

Megjegyzés:

Sajnálatos módon az atlasz kiszámíthatatlan viselkedést produkált a fejlesztés alatt.