

Morze dekodoló programozói specifikáció

Fájlok

- `kodtabla.txt` : ez a fájl tartalmazza a morze kódokat és azoknak betű megfelelőit
- `kod.txt`: ez a fájl csupán tesztelési okok miatt van hozzárakva ha ki akarjuk próbálni hogyan reagál a program ha egy hiányos vagy másik kódtáblát adunk meg

Struktúrák

a struktúra a `tree.h` ban van definiálva

- `tree`: a dekodoló fa ideiglenes tárolására szolgál:
 - `data[20]` és `data2[20]` a morze és a betű stringek
 - `tree.left` és `tree.right` a fa ágainak saját struktúra alapján definiált pointerrei

A bináris fa dinamikusan van foglalva attól függően hogy a fájlban mennyi adat található. Minden alkalommal amikor új fájl szerelnénk beolvasni a fa fel lesz szabadítva és annak helyére újra lesz foglalva a kért fájl adata.

Modulok

- `file_read.c`: a fájl beolvasásával foglalkozik és egy teszt függvénnyel
 - függvényei:
 - `file_test` függvény egy string-et kér ami a megnyitandó fájl helye, ha sikerül meghívnia akkor 1-et ad vissza értékként ami jelzi a mainben hogy sikeres volt a fájl meghívása tehát nem kell visszaállítani az eredeti fájlra a forrást
 - `thead` függvény egy `tree` típusú pointert kér és egy fájl címet string alakban, ezek után ha tudja akkor beolvassa a fájlt és feldolgozza az adatokat a `tree_append` függvény segítségével. Ha nem tudja megnyitni a fájlt akkor megpróbálja az eredeti forrást használni de ha az sem sikerül akkor kiírja hogy hiba lépett fel és kilép a programból miután hiányzik egy alap elem a program futásához

A függvényben foglalt a és b változók a fájl beolvasásánál szükségesek mert ideiglenesen ebben tárolódnak az adatok, ezek a függvény végén fel vannak szabadítva

A függvény egy `tree` típusú pointerrel tér vissza ami a gyökere az egész dekodoló fára
- `tree.c`: a dekodoló fa kezelésével és feldolgozásával foglalkozik
 - függvényei:
 - `new_branch` függvény egy `tree` típusú pointert kér és ezzel is fog visszatérni csak lefoglalja és kiegészíti a fát egy álfával
 - `tree_append` függvény egy `tree` típusú pointert kér, 3 db stringet és egy `int` típusú változót ami a függvény mélységét határozza meg. Először ha a megadott fa üres teljesen akkor létrehoz a `new_branch` függvény segítségével egy új álfát és tovább megy majd ha a data azaz a morze kód soron következő betűje nem a lezáró nulla akkor megvizsgálja hogy melyik irányba menjen, ha – akkor balra ha . akkor jobbra és ez alapján feltölti a fát a soron következő elemekkel.
 - `ftree` függvény egy fa felszabadító függvény ami minden alkalommal mikor új fájlból akarunk beolvasni vagy a program a végéhez ér felszabadítja az eddig használt és lefoglalt fa területét
- `decoder.c`: a szövegek dekódolásával foglalkozik
 - függvényei:
 - `decode` függvény a morze szöveget alakítja át sima abc-re a dekodoló fa segítségével, egy `tree` típusú pointert kér ami alapján megkeresi benne az egyes morze megfelelőjét. A függvény beolvas egy szöveget ami maximum 255 hosszú lehet és létrehoz egy `char_err` változót ami a függvény végén fogja megmondani ha volt olyan karakter amit nem talált meg a fában és ezt egy paragrafus jellel jelzi a lefordított szövegben is. Megvizsgálja karakterenként a morze kódot és ha – vagy . a következő akkor a megfelelő irányban fog lefele haladni a fában majd kiírja a keresett karaktert ha megtalálta.

- `encode` függvény az abc betűit fogja megkeresni és kiírni, paraméterként egy tree típusú pointert kér. A függvény bekéri a felhasználótól a maximum 255 hosszú szöveget majd a `find` függvény segítségével kiírja a megtalálendő karakter morze kódját.
- `find` függvény egy adott karaktert kér paraméternek és egy tree típusú pointert majd ezek után egy mélységi bejárást alkalmazva megtalálja a keresett betűt és annak morze megfelelőjét (bejárja az egész fát)
- `main.c`: a menüt kezeli és a választott funkciót hívja meg és dolgozza fel
 - létrehoz egy `int` típusú `b` változót ami a felhasználói bemenettel együtt változik hogy ne kelljen a felhasználónak számokat használnia hanem beírhatta a parancsokat
 - létrehoz egy 256 hosszú stringet amiben tárolja a kiválasztott fájlt amiből táplálkozik a program, alap értéke a `kodtabla.txt`
 - kiírja a menü opcióit minden indításkor először
 - egy hátul tesztelési ciklus bekérdezi a parancsot és a bemenet alapján eldönti hogy melyik almenü legyen aktív
 - a bemenettől függően meghívható a dekódolás, kódolás, fájl váltás, help menü és a kilépés a programból
 - a függvény után az `ftree` segítségével felszabadítja az utoljára használt fát