Név: Nagy Róbert

Neptun kód: JMDRGG

1.feladat: A system() rendszerhívással hajtson végre létező és nem létező parancsot, és vizsgálja a visszatérési érteket!

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>

int main(int argc, char const *argv[]) {

    int x = system("ls");

    if(WIFEXITED(x)) {
        printf("Kilepett\n");
        printf("Visszateresi ertek: %d\n", WEXITSTATUS(x));
    }

    int y = system("ip"); // nem létező parancs

    if(WIFEXITED(y)) {
        printf("Kilepett\n");
        printf("Visszateresi ertek: %d\n", WEXITSTATUS(y));
    }

    return 0;
}
```

A system(const *char parancs) utasítás visszatérési értéke -1, ha a megadott parancs helytelen, egyébként a parancs visszatérési értéke. A system() utasítás a stdio.h headerben található.

A WIFEXITED(status) értéke igaz, ha normális leállása volt a processznak, egyébként hamis.

WEXITSTATUS(status) visszatérési értéke a parancs visszatérése.

2. Írjon programot, amely billentyűzetről bekér Unix parancsokat és végrehajtja őket, majd kiírja a szabványos kimenetre.

```c
int main(int argc, char const *argv[])
{
    char parancs[50];
    printf("Adj meg egy parancsot: ");
    scanf("%s", parancs);

    int x = system(parancs);

    if(WIFEXITED(x))
    {
        printf("Kilepett\n");
    }
    WEXITSTATUS(x) ? printf("Hibas mukodes \n") : printf("Normalis mukodes ");

    return 0;
}
```

3. Készítsen egy parent.c és egy child.c programokat. A parent.c elindít egy gyermek processzt, ami különbözik a szülőtől. A szülő megvárja a gyermek lefutását. A gyermek szöveget ír a szabványos kimenetre

A fork() parancs gyerek processz létrehozását teszi lehetővé. A függvény negatív értékkel tér vissza, ha nem sikerül gyerek processzt létrehozni, 0 értékkel tér vissza, ha sikeres, pozitív értékkel tér vissza, ha a gyerek process visszatér a szülő processzhez.

Az execl() parancs változó argumentum listát használ. Az első argumentum a parancs relatív ösvénye. Az ezt követő argumentumok pedig a megadott parancs argumentumai. A parancs első argumentuma a parancs neve. Az utolsó paraméter pedig lezáró NULL. A függvény normális működés esetén nem ad vissza értéket, egyébként -1 et ad vissza.

A waitpid() utasítás 3 paraméterrel rendelkezik. Az első parameter a process id. Második parameter egy int* változó, ami a státuszt fogja tárolni. A harmadik parameter pedig egy opció.

Visszatérési értéke a gyerek process pid-je, normális esetben, ha nem létező pid akkor 0, ha hibás, akkor -1.

Az operácós parameter lehet < -1, ekkor ha a gyerek processz gpid-je megegyezik a valódi pid-el, akkor várja meg. Lehet -1, ekkor várjon meg bármely gyerek processzt. Lehet 0, ekkor ha a gyerek processz gpid-je megegyezik a pid-el, akkor várja meg. Ha > 0, akkor várja meg azt a gyerek processzt, amelynek a pid-je megegyezik a megadott pid-el.

4. A fork() rendszerhívással hozzon létre egy gyerek processzt-t és abban hívjon meg egy exec családbeli rendszerhívást (pl. execlp) egy unix-paranccsal. A szülő várja meg a gyerek futását!

5. A fork() rendszerhívással hozzon létre gyerekeket, várja meg és vizsgálja a befejeződési állapotokat (gyerekben: exit, abort, nullával való osztás)!