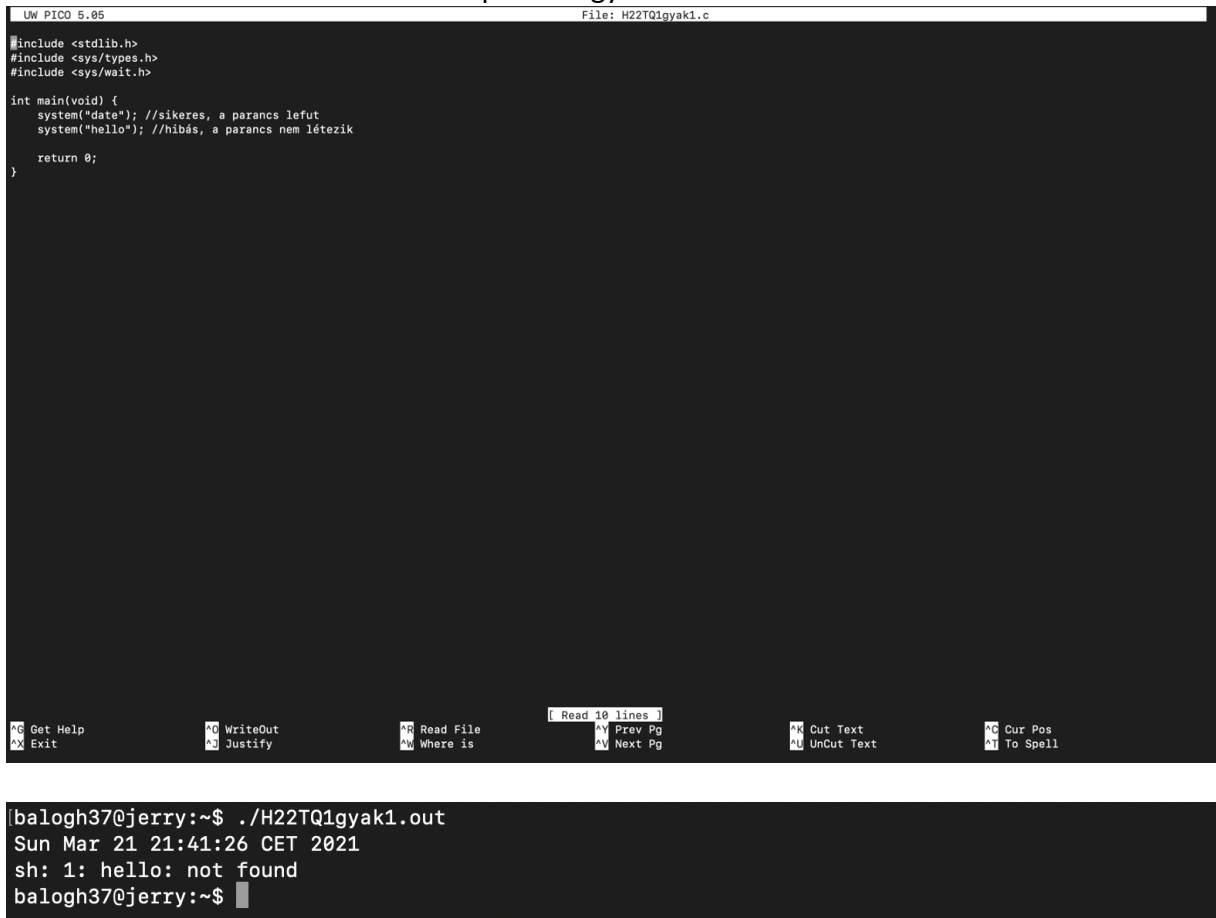


1. A `system()` rendszerhívással hajtson végre létező és nem létező parancsot, és vizsgálja a visszatérési értéket! Mentés: `neptunkodgyak1.c`



The image shows a code editor window titled "UW PICO 5.05" and "File: H22TQ1gyak1.c". The code is as follows:

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

int main(void) {
    system("date"); //sikeres, a parancs lefut
    system("hello"); //hibás, a parancs nem létezik
    return 0;
}
```

Below the editor, the execution output is shown in a terminal window:

```
balogh37@jerry:~$ ./H22TQ1gyak1.out
Sun Mar 21 21:41:26 CET 2021
sh: 1: hello: not found
balogh37@jerry:~$
```

2. Írjon programot, amely billentyűzetről bekér Unix parancsokat és végrehajtja őket, majd kiírja a szabványos kimenetre. (pl.: amit bekér: `date`, `pwd`, `who` etc.; kilépés: `CTRL-\`) Mentés: `neptunkodgyak2.c`

```
Uw PICO 5.05                                     File: H22TQ1gyak2.c                               Modified

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

#define BUFFER_SIZE 255

int main(void) {
    char buffer[BUFFER_SIZE];
    do
    {
        printf(">");
        fgets(buffer, BUFFER_SIZE, stdin);
        system(buffer);
    }
    while (1);

    return 0;
}

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Pg      ^K Cut Text      ^C Cur Pos
^X Exit          ^N Justify       ^W Where is      ^V Next Pg      ^U UnCut Text    ^T To Spell

balogh37@jerry:~$ ./H22TQ1gyak2.out
>
[>date
Sun Mar 21 21:47:38 CET 2021
[>who
balogh37 pts/0          2021-03-21 21:33 (2a00:1110:200:df2b:60b5:cb47:b2a:8963)
[>pwd
/home/stud2019/balogh37
```

3. Készítsen egy parent.c és a 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 (5-ször) (pl. a hallgató neve és a neptunkód)! Mentés: parent.c, ill. child.c

```
UW PICO 5.05                                     File: child.c                               Modified
#include <stdio.h>
#include <unistd.h>

int main(void) {
    for (int i = 0; i < 5; i++)
    {
        printf("Balogh Bianka \n H22TQ1\n");
        sleep(1);
    }
    return 0;
}
```

AG Get Help AC WriteOut AR Read File AY Prev Pg AX Cut Text AC Cur Pos  
AX Exit AD Justify AW Where is AV Next Pg AU UnCut Text AT To Spell

```
UW PICO 5.05                                     File: parent.c                               Modified
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>

int main(void) {
    int pid;

    if ((pid = fork()) < 0)
        perror("fork error");
    else if (pid == 0)
    {
        if (execl("./child", "child", (char *)NULL) < 0)
            perror("execl error");
    }
    if (waitpid(pid, NULL, 0) < 0)
        perror("wait error");

    return 0;
}
```

AG Get Help AC WriteOut AR Read File AY Prev Pg AX Cut Text AC Cur Pos  
AX Exit AD Justify AW Where is AV Next Pg AU UnCut Text AT To Spell

1221Q1

```
balogh37@jerry:~$ ./child.out
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
```

4. A fork() rendszerhívással hozzon létre egy gyerek processzt-t és abban hívjon meg egy exec családbeli rendszerhívást (pl. execlp). A szülő várja meg a gyerek futását! Mentés: neptunkodgyak4.c

```
UW PICO 5.05 File: H22TQ1gyak4.c Modified
{
    int pid;
    int status;

    if ((pid = fork()) < 0)
        perror("fork hiba");
    else if (pid == 0)
        if (execl("./child.out", "child", (char *)NULL) < 0)
            perror("execl error");
    if (wait(&status) != pid)
        perror("wait hiba");

    return 0;
}
```

Get Help WriteOut Read File Prev Pg Cut Text Cur Pos  
Exit Justify Where is Next Pg UnCut Text To Spell

```
balogh37@jerry:~$ ./H22TQ1gyak4.out
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
Balogh Bianka
H22TQ1
```

5. A fork() rendszerhívással hozzon létre gyerekeket, várja meg és vizsgálja a befejeződési állapotokat (gyerekekben: exit, abort, nullával való osztás)! Mentés: neptunkodgyak5.c

```
UW PICO 5.05 File: H22TQ1gyak5.c Modified
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int pid;
    int status;

    if ((pid = fork()) < 0)
        perror("fork hiba");
    else if (pid == 0)
        exit(6);
    if (wait(&status) != pid)
        perror("wait hiba");
    if (WIFEXITED(status))
        printf("Befejeződés normális, visszaadott érték: %d\n", WEXITSTATUS(status));

    return 0;
}
```

AG Get Help    W WriteOut    AR Read File    AY Prev Pg    AK Cut Text    AG Cur Pos  
EX Exit        J Justify        W Where is    AV Next Pg    AL UnCut Text    AT To Spell

```
[balogh37@jerry:~$ ./H22TQ1gyak5.out
Befejeződés normális, visszaadott érték: 6
```