

Operációs Rendszerek BSc

5. Gyak.

2022.03.13.

Készítette:

Tóth Dániel Márk BSc

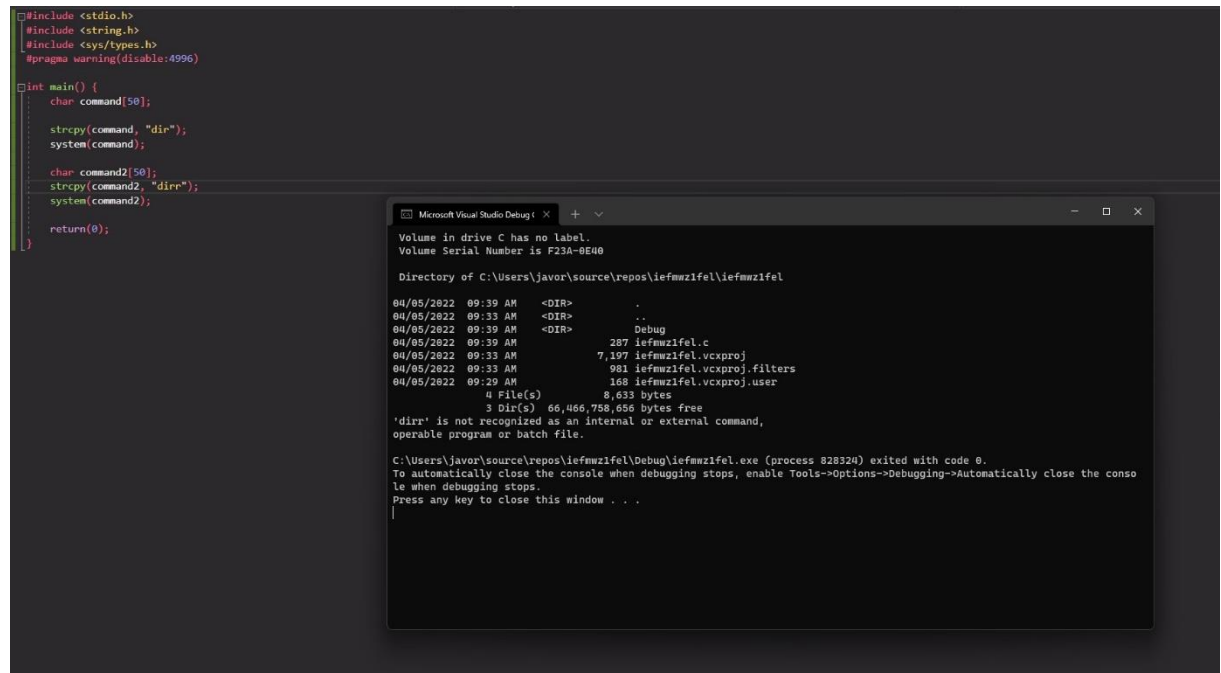
Mérnökinformatika

IEFMWZ

Miskolc, 2022

1. A `system()` rendszerhívással hajtson végre létező és nem létező parancsot, és vizsgálja a visszatérési értéket, magyarázza egy-egy mondattal
A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.
Mentés: `neptunkod1fel.c`

A képen látható C kóddal meghívtam a Linux „`dir`” parancsát. A program sikeresen lefutott. Míg a kód másik részében pedig a „`dirr`” parancsot kértem be, ami tudni illik nem létezik, így egy olyan hiba üzenetet dobott ki a program, ami a képen látható.



```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#pragma warning(disable:4996)

int main() {
    char command[50];
    strcpy(command, "dir");
    system(command);

    char command2[50];
    strcpy(command2, "dirr");
    system(command2);

    return(0);
}
```

Microsoft Visual Studio Debug Console

Volume in drive C has no label.
Volume Serial Number is F23A-8E40

Directory of C:\Users\javor\source\repos\iefmwz1fel\iefmwz1fel

04/05/2022 09:39 AM	<DIR>	.
04/05/2022 09:33 AM	<DIR>	..
04/05/2022 09:39 AM	<DIR>	Debug
04/05/2022 09:39 AM		287 iefmwz1fel.c
04/05/2022 09:33 AM		7,197 iefmwz1fel.vcxproj
04/05/2022 09:33 AM		981 iefmwz1fel.vcxproj.filters
04/05/2022 09:29 AM		168 iefmwz1fel.vcxproj.user
	4 File(s)	8,633 bytes
	3 Dir(s)	66,466,788,658 bytes free

'dirr' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\javor\source\repos\iefmwz1fel\Debug\iefmwz1fel.exe (process 828324) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

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-\\) - magyarázza egy-egy mondattal
A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.
Mentés: neptunkod2fel.c

A feladatot hasonlóra csináltam meg, mint az elsőt azzal a különbséggel, hogy felhasználótól kérem be a parancsot. Bármit beír a konzol ablakba, valamiféle Linux parancsot fog meghívni, vagy esetlegesen hiba üzenetet. Itt a „dir” parancsot írtam be és az eredmény látható a lenti képen.

The image shows a screenshot of a C program in Visual Studio and its console output. The program, named `neptunkod2fel.c`, is designed to read a Unix command from the user and execute it. The code includes `<stdio.h>`, `<string.h>`, and `<sys/types.h>`, and uses `scanf` to read a command into a buffer, then `strcpy` to copy it and `system` to execute it. The console window shows the output of the `dir` command, displaying the directory structure of `C:\Users\javor\source\repos\iefmwz1fel\iefmwz1fel`. The output includes the volume information for drive C, the directory listing, and the exit status of the process.

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#pragma warning(disable:4996)

int main() {
    char command[50];
    char command2[50];
    scanf("%s", command2);
    strcpy(command, command2);
    system(command);

    return(0);
}
```

Microsoft Visual Studio Debug | x + -

dir
Volume in drive C has no label.
Volume Serial Number is F23A-0E40

Directory of C:\Users\javor\source\repos\iefmwz1fel\iefmwz1fel

04/05/2022	09:46 AM	<DIR>	.
04/05/2022	09:33 AM	<DIR>	..
04/05/2022	09:46 AM	<DIR>	Debug
04/05/2022	09:41 AM		7,197 iefmwz1fel.vcxproj
04/05/2022	09:41 AM		981 iefmwz1fel.vcxproj.filters
04/05/2022	09:29 AM		168 iefmwz1fel.vcxproj.user
04/05/2022	09:46 AM		260 iefmwz2fel.c
		4 File(s)	8,606 bytes
		3 Dir(s)	66,450,581,632 bytes free

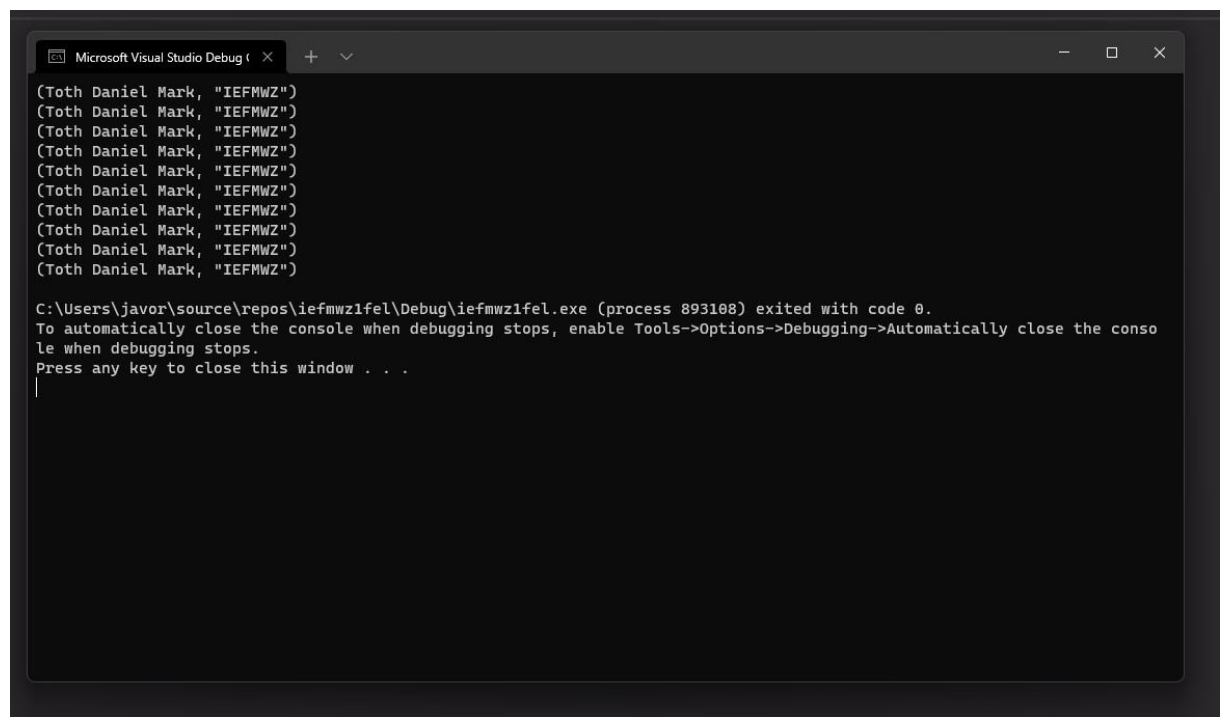
C:\Users\javor\source\repos\iefmwz1fel\Debug\iefmwz1fel.exe (process 861060) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

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 (10-ször) (pl. a hallgató neve és a neptunkód)! - magyarázza egy-egy mondattal

A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.

Mentés: parent.c, ill. child.c

Itt megcsináltam a két .c fájlt, mellé egy összekötő .h header file-t. A gyermek programba illesztettem azt a metódust, ami kiírja a képernyőre az adott szöveget 10-szer. Majd ezt a szülő program meghívja és sikeresen futtatja. A végeredmény a képen látható.



```
Microsoft Visual Studio Debug Console
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")
(Toth Daniel Mark, "IEFMWZ")

C:\Users\javor\source\repos\iefmwz1fel\Debug\iefmwz1fel.exe (process 893108) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

A feladatok többi részénél nem sikerült képernyőképet készítenem technikai okok miatt, de a programok, amiket írtam sikeresen lefutottak és a várt eredményt hozták!

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! - magyarázza egy-egy mondattal.
Mentés: neptunkod4fel.c

Ebben a programban a leírtak szerint meghívtam a fork() rendszerhívást, majd létrehoztam egy gyermek processzt, ami a háttérben lefutott.

5. A fork() rendszerhívással hozzon létre gyerekeket, várja meg és vizsgálja a befejeződési állapotokat (gyerekben: exit, abort, nullával való osztás)! - magyarázza egy-egy mondattal!
A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.
Mentés: neptunkod5fel.c

Ebben a feladatban sikeresen létrehoztam a fork() rendszerhívással az exit, abort metódusokat, majd megírtam a nullával való osztás metódusát is, ami szintén gyerek metódusként fut le.

6.
Adott a következő ütemezési feladat, amit a FCFS, SJF és Round Robin (RR) ütemezési algoritmus használatával készítsen el (külön-külön táblázatba):
I. Határozza meg FCFS és SJF esetén
a.) A befejezési időt?
b.) A várakozási/átlagos várakozási időt?
c.) Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét.
Megj.: a Gantt diagram ábrázolása szerkesztő program segítségével vagy Excel programmal.

FCFS esetén:

FCFS	P1	P2	P3	P4	P5
Érkezés	0	1	3	9	12
CPU Idő	3	8	2	20	5
Indulás	3	11	13	33	38
Befejezés	3	10	10	24	26
Várakozás	0	2	8	4	21

Átlagos várakozási idő: 7

SJF esetén:

SJF	P1	P2	P3	P4	P5
Érkezés	0	1	3	9	12
CPU Idő	3	8	2	20	5
Indulás	3	13	5	38	18
Befejezés	3	12	2	29	6
Várakozás	0	4	0	9	1

Átlagos várakozási idő: 2.8

RR, 5ms esetén:

SJF	P1	P2	P3	P4	P5
Érkezés	0	1	3	9	12
CPU Idő	3	8	2	20	5
Indulás	3	13	10	38	23
Befejezés	3	12	7	29	11
Várakozás	0	4	5	9	6

Átlagos várakozási idő: 4.8

Végrehajtási sorrend:

P1-P2-P3-P2-P4-P5-P4-P4-P4