

Jegyzőkönyv

Operációs rendszerek BSc

2022.Tavaszi féléves feladat.

Készítette:

Bereznai Benjamin BSc

Mérnök informatikus

CVVJZ4

Miskolc, 2022

1. feladat

13. Adott négy processz (A, B, C, D) a rendszerbe, induláskor a p_cpu értéke A=0, B=0, C=0, D=0. A rendszerben a P_USER = 60. Az óráütés 1 indul, a befejezés 301-ig.

Induláskor a p_usrpri A=60, B=65, C=60 és D=60.

Induláskor a p_nice értéke A=0, B=5, C=0 és D=0.

a.) Határozza meg az ütemezést RR 301 óráütésig - táblázatba!

b.) Minden óráütem esetén határozza meg a processzek sorrendjét óráütés előtt/után.

c.) Igazolja a számítással a tanultak alapján.



A kezdeti feltételek fontos h a B processz p_nice értéke 5 ezért azt kerül a prioritási lista utolsó helyére.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	A proc		B proc		C proc		D proc		Reschedule								
2	Óra ütem	p_usrpri	p_cpu	p_usrpri	p_cpu	p_usrpri	p_cpu	p_usrpri	p_cpu	Run_before	Run_after			p_nice			
3	Start point	60	0	65	0	60	0	60	0	0	A			A	0		p_usrpri=p_usrpri+p_cpu/2+2*p_nice
4	1	60	1	65	0	60	0	60	0	A	A			B	5		p_cpu=p_cpu*KF
5	2	60	2	65	0	60	0	60	0	A	A			C	0		
6	3	60	3	65	0	60	0	60	0	A	A			D	0		(Vincze Jegyzet)
7	4	60	4	65	0	60	0	60	0	A	A						
8	5	60	5	65	0	60	0	60	0	A	A			p_usrpri			
9	6	60	6	65	0	60	0	60	0	A	A			A	60		
10	7	60	7	65	0	60	0	60	0	A	A			B	65		
11	8	60	8	65	0	60	0	60	0	A	A			C	60		
12	9	60	9	65	0	60	0	60	0	A	A			D	60		
13	10	60	10	65	0	60	0	60	0	A	C						
14	11	60	10	65	0	60	1	60	0	C	C			P_user	60		
15	12	60	10	65	0	60	2	60	0	C	C						
16	13	60	10	65	0	60	3	60	0	C	C			KF	0,75		

A KF értéke egy konstans szám 0.75 (Vincze Dávid jegyzet)

A két alap képlet amivel dolgozunk:

- $p_cpu = p_cpu * KF$
- $p_usrpri = p_user + (p_cpu/2) + (2 * p_nice)$

98	95	60	30	65	20	60	26	60	20	C	C						
99	96	60	30	65	20	60	27	60	20	C	C						
100	97	60	30	65	20	60	28	60	20	C	C						
101	98	60	30	65	20	60	29	60	20	C	C						
102	99	60	30	65	20	60	30	60	20	C	C						
103	100	71	22	86	22	67	15	67	15	B	D			D futott legrégebben			
104	101	71	22	86	22	67	15	67	16	D	D						
105	102	71	22	86	22	67	15	67	17	D	D						
106	103	71	22	86	22	67	15	67	18	D	D						
107	104	71	22	86	22	67	15	67	19	D	D						
108	105	71	22	86	22	67	15	67	20	D	D						
109	106	71	22	86	22	67	15	67	21	D	D						
110	107	71	22	86	22	67	15	67	22	D	D						

A 100. óráütemnél meghatározzuk a legrégebben futott processzt a fent megadott képletek alapján.

200	197	71	22	86	22	67	15	67	112	D	D						
201	198	71	22	86	22	67	15	67	113	D	D						
202	199	71	22	86	22	67	15	67	114	D	D						
203	200	68	16	83	16	65	11	102	85	D	A	A futott legrégebben					
204	201	68	17	83	16	102	84	65	11	A	A						
205	202	68	18	83	16	102	84	65	11	A	A						
206	203	68	19	83	16	102	84	65	11	A	A						

Szintén így járunk el a 200. óraütemnél is.

301	298	68	114	83	16	102	84	65	11	A	A						
302	299	68	115	83	16	102	84	65	11	A	A						
303	300	103	86	81	12	91	63	64	8	A	B	futott régebben					
304	301	103	86	81	13	91	63	64	8	B							
305																	

Majd a 300. óra ütemet követően is.

2.feladat

16. Írjon egy olyan C programot, mely egy fájlból számpárokat kiolvassa meghatározza a legnagyobb közös osztóját. A feladat megoldása során használjon nevesített csővezetékot, valamint a kimenet kerüljön egy másik fájlba. A kimeneti fájl struktúrája kötött!

Példa a bemeneti és kimeneti fájl struktúrájára:

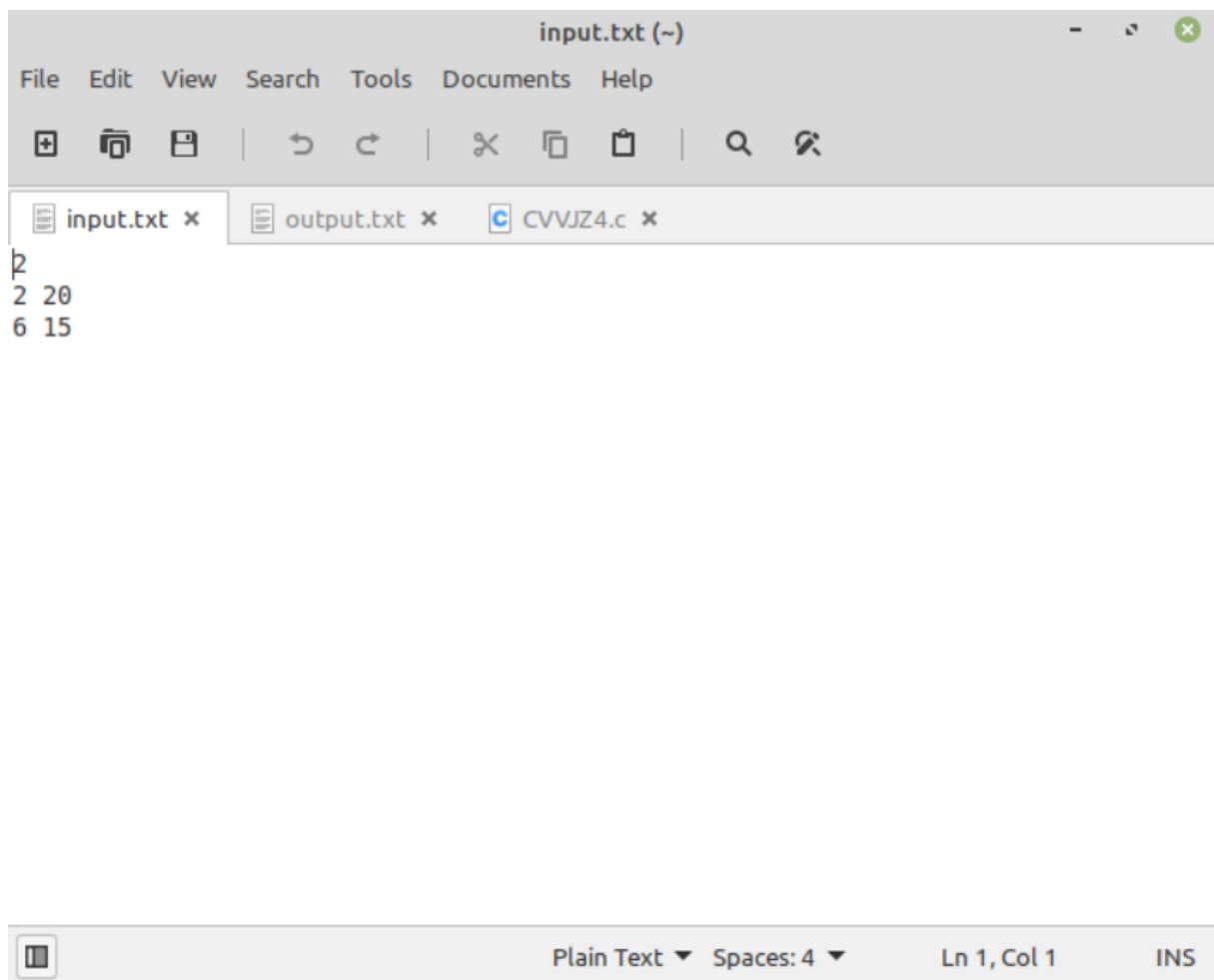
Bemeneti fájl:

i (Ez jelzi a számpárok darabszámát)
x y

Kimeneti fájl(Az x,y jelzi a bemeneti adatokat a z pedig a kimenet eredményét):
x y z

1. Bemeneti file létrehozása. Megnyitása első sor kiolvasása a file ból ami meghatározza hogy hány osztópárunk lesz.
2. Beolvasni az osztópárokat.
3. Meghatározzuk az LNKO algoritmus segítségével a legnagyobb közös osztókat.
4. Tömb nullázó függvényeket alkalmatunk szemét elkerülése érdekében.
5. Ellenőrzötten megnyitjuk az output.txt -t megnézzük hogy megnyílik e és hogy létre van e hozva.
6. Létre hozunk egy buffert az input.txt tartalmának.
7. Meghatározzuk a számpárok darabszámát.
8. Meghatározzuk a szám karaktereinek darabszamat
:floor(log10(abs(INTEGER_VARIABLE)))+1
9. Létrehozunk egy csővezetékot.
- 10.pid_t és fork() segítségével létrehozuk a gyerekprocesszt.
- 11.Kiírunk az OUTPUT.txt be
- 12.Bezárjuk és letöröljük a csővezetékot.

Input File tartalma :

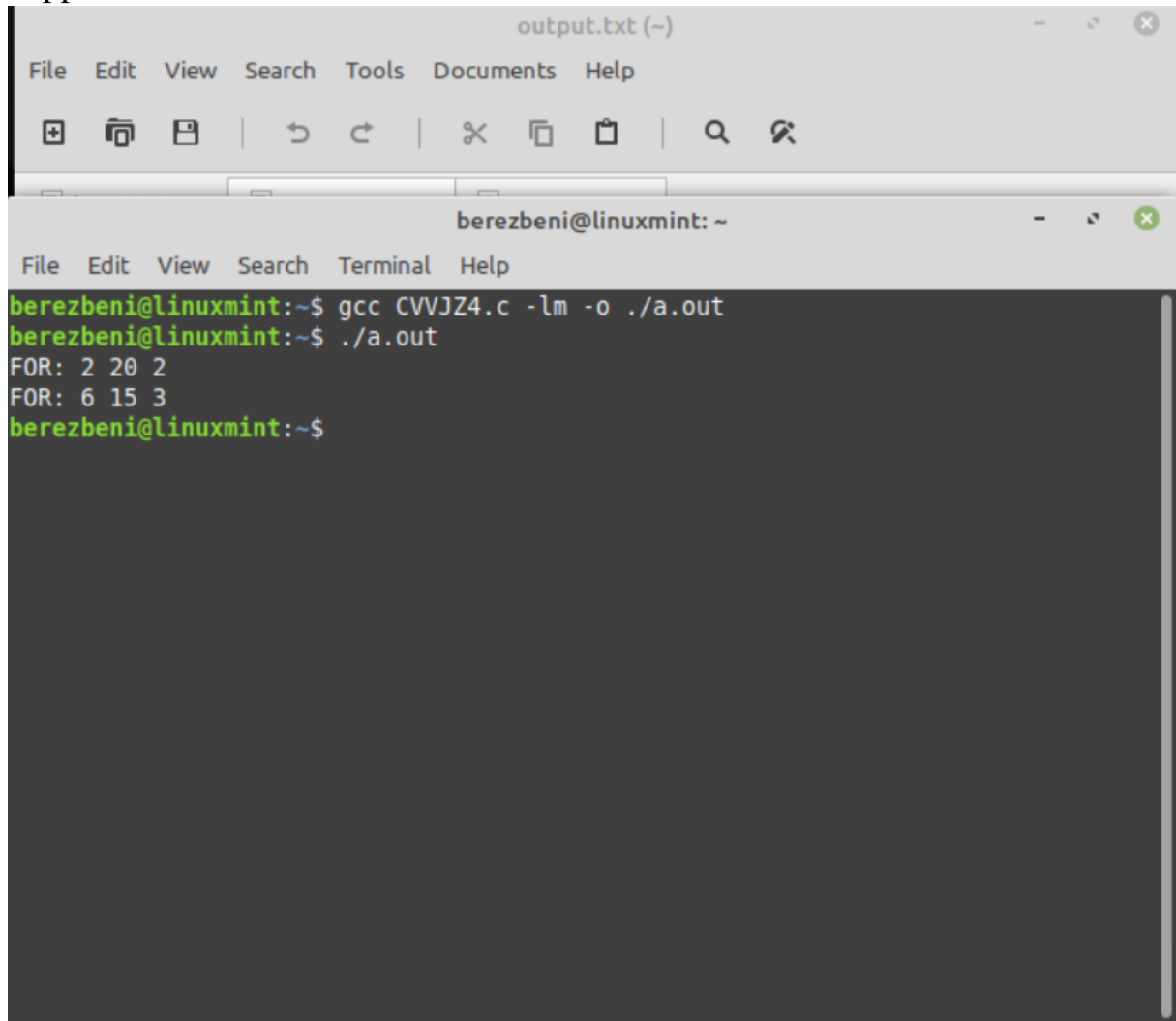


The screenshot shows a text editor window with the title "input.txt (~)". The menu bar includes "File", "Edit", "View", "Search", "Tools", "Documents", and "Help". The toolbar contains icons for adding files, saving, undo, redo, cut, copy, paste, search, and a cursor icon. The tab bar shows three open files: "input.txt", "output.txt", and "CVVJZ4.c". The main text area contains the following content:

```
2  
2 20  
6 15
```

The status bar at the bottom indicates the file is in "Plain Text" format with "Spaces: 4", and the cursor is at "Ln 1, Col 1" in "INS" mode.

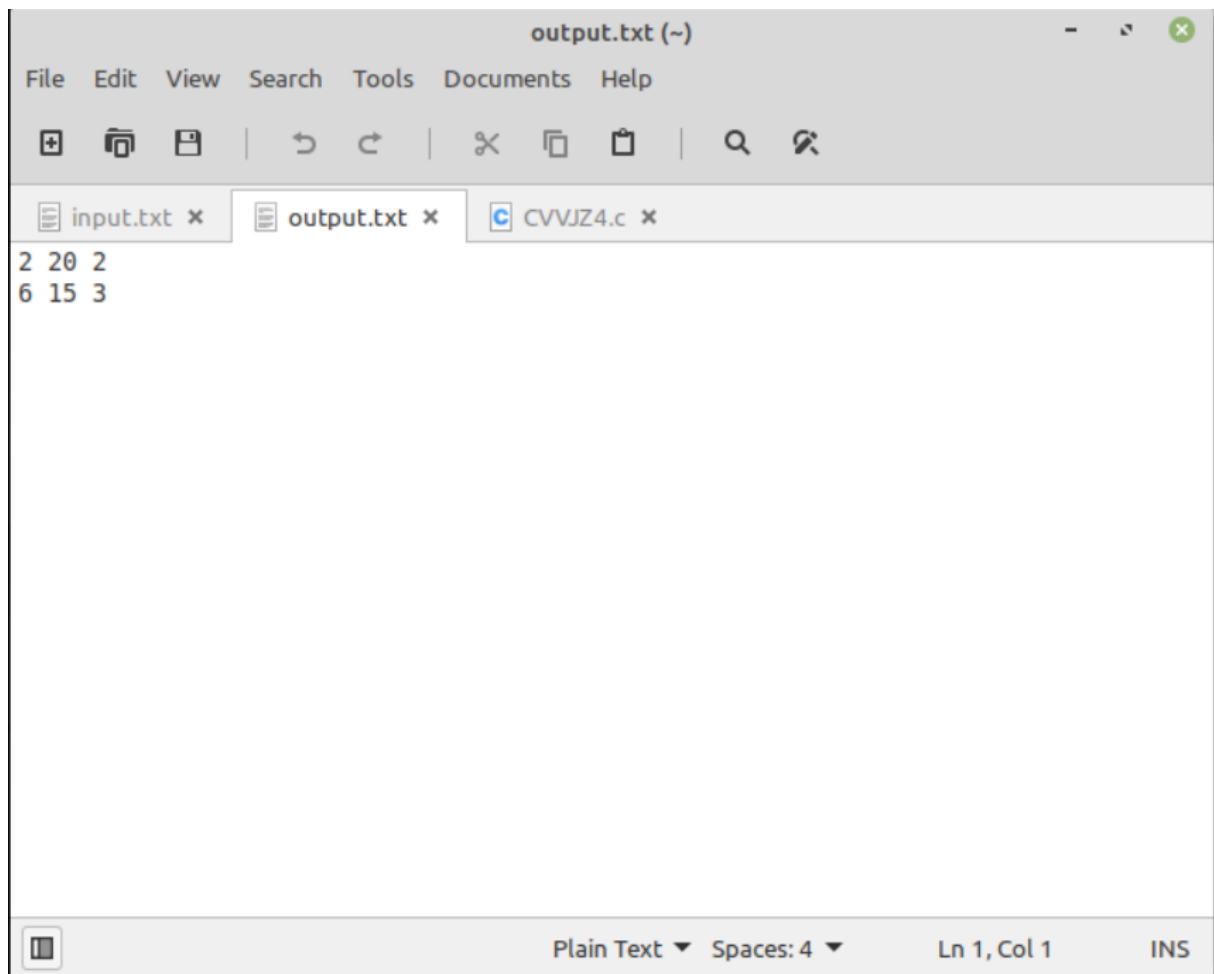
A futó program megnyitása és az eredmények kiírva konzolra ellenőrzés képpen!



The image shows a screenshot of a Linux terminal window. The window title is "output.txt (~)". The menu bar includes "File", "Edit", "View", "Search", "Tools", "Documents", and "Help". The toolbar contains icons for file operations and search. The terminal window title is "berezbeni@linuxmint: ~". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal output shows the following commands and results:

```
berezbeni@linuxmint:~$ gcc CVVJZ4.c -lm -o ./a.out
berezbeni@linuxmint:~$ ./a.out
FOR: 2 20 2
FOR: 6 15 3
berezbeni@linuxmint:~$
```

Az Output file tartalma:



The screenshot shows a text editor window titled "output.txt (~)". The window has a menu bar with "File", "Edit", "View", "Search", "Tools", "Documents", and "Help". Below the menu bar is a toolbar with icons for adding, opening, saving, undo, redo, cut, copy, paste, find, and replace. The editor has three tabs: "input.txt x", "output.txt x", and "CVVJZ4.c x". The "output.txt" tab is active, showing the following text:

```
2 20 2  
6 15 3
```

The status bar at the bottom indicates "Plain Text", "Spaces: 4", "Ln 1, Col 1", and "INS".