

Operációs rendszerek BSc

7.gyak.

2021. 03. 24.

Készítette:

Kacsir András

VSG9L4

Programtervező informatikus

Miskolc, 2021.03.24.

1. feladat – Adott négy processz a rendszerbe, melynek beérkezési sorrendje: A, B, C és D. Minden

processz USER módban fut és mindegyik processz futásra kész.

Kezdetben mindegyik processz $p_uspri = 60$.

Az A, B, C processz $p_nice = 0$, a D processz $p_nice = 5$.

Mindegyik processz $p_cpu = 0$, az óráütés 1 indul, a befejezés legyen

201. óraütés-ig.

a.) Határozza meg az ütemezést RR nélkül és az ütemezést RR-nal - külön-külön táblázatba.

b.) Minden óraütem esetén határozza meg a processzek sorrendjét óraütés előtt/után.

c.) Igazolja a számítással a tanultak alapján.

$KF = 2 \cdot FK / (2 \cdot FK + 1)$ - korrekciós faktor;

$p_cpu = p_cpu \cdot KF$, ahol KF értéke 1/2;

$p_pri = P_USER + p_cpu / 4 + 2 \cdot p_nice$;

A táblázat javasolt formája RR/RR nélkül a következő:

	A process		B process		C process		D process		Reschedule	
Clock tick	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_uspri	p_uspri	p_cpu	running before	running after
Starting point	60	0	60	0	60	0	60	0		
1
...

	A process		B process		C process		D process		Reschedule		
Clock tick	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	p_uspri	p_cpu	running before	running after	P_USER
Starting point		60	0	60	0	60	0	60	0	A	60
1		60	1	60	0	60	0	60	0	A	
...	A	A	
99		60	99	60	0	60	0	60	0	A	
100		73	50	60	0	60	0	60	0	A	
101		73	50	60	1	60	0	60	0	B	
...	B	B	
199		73	50	60	99	60	0	60	0	B	
200		66	25	73	50	60	0	60	0	B	
201		66	25	73	50	60	1	60	0	C	
...	C	C	
299		66	25	73	50	60	99	60	0	C	
300		63	13	66	25	73	50	60	0	C	
301		63	13	66	25	73	50	60	1	D	
...	D	D	
399		63	13	66	25	73	50	60	99	D	
400		62	7	63	13	66	25	83	50	D	

2. feladat -A tanult rendszer hívásokkal (open(), read()/write(), close() - ők fogják a rendszerhívásokat tovább hívni.) írjanak egy neptunkod_openclose.c programot, amely megnyit egy fájlt – neptunkod.txt, tartalma: hallgató neve, szak , neptunkod.

A program következő műveleteket végezze:

- olvassa be a neptunkod.txt fájlt, melynek attribútuma: O_RDONLY
- hiba ellenőrzést,
- write() - mennyit ír ki a konzolra.
- read() - kiolvassa a neptunkod.txt tartalmát és mennyit olvasott ki (byte), és kiírja konzolra.
- lseek() – pozícionálja a fájl kurzor helyét, ez legyen a fájl eleje: SEEK_SET, és kiírja a konzolra

```
Megnyitottam a(z) "VSG9L4.txt" fájlt!  
Beolvasott tartalom: "Kacsir Andras, Programtervező informatikus, VSG9L4" összesen "50" byte.  
A fájl elejére állítottuk a mutatót.  
A fájlba írtuk a(z) "Rendszerhívással írás fájlba" szöveget. összesen "29" byte.  
  
C:\Users\Andrew\source\repos\VSG9L4_openclose\Debug\VSG9L4_openclose.exe (process 7188) 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 . . .
```

Írjanak magyarázatot a jegyzőkönyvbe.

További információk: man 2 open; man 2 read; man 2 write; man 2 close.

Open megnyitja a fájlt

read olvassa a fájlt

write fájlba ír

close bezárja a fájlt