Operációs rendszerek – 5. Gyakorlat

Linux OS - Rendszerhívások, processz ütemezés

Töltse fel az aktuális mappába: **Neptunkod_....**

Jegyzőkönyv neve: *neptunkodgya5.pdf*

Forrás file-k

Határidő: 2021.03.10. ill. módosítás esetén 2021.03.14.

Feladatok

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

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*

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*

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*

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

Mentés: *neptunkodgyak5.c*

6. Adott a következő ütemezési feladat, ahol a RR ütemezési algoritmus használatával készítse el: Határozza meg a

a.) Ütemezze az adott időszelet alapján az egyes processzek paramétereit (ms)!

b.) A rendszerben lévő processzek végrehajtásának sorrendjét?

c.) Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét!

RR: 5 ms	**Round Robin**				
	P1	**P2**	**P3**	**P4**	**P5**
Érkezés	0	1	3	9	12
CPU idő	3	8	2	20	5
Indulás	0	3	8	13	18
Befejezés					
Várakozás					