

# Operációs rendszerek BSc

## 9. gyakorlat

### **Készítette:**

Nemesi Gergely Tibor  
Üzemmérnök-informatikus  
Neptun: ILZGJC

**2022.04.04**

## Contents

<b>I IPC mechanizmus - Szignálkezelés</b>	<b>2</b>
<b>1. feladat</b> . . . . .	<b>2</b>
<b>2. feladat</b> . . . . .	<b>3</b>
A, rész . . . . .	3
B, rész . . . . .	3
C, rész . . . . .	3
D, rész . . . . .	3
<b>3. feladat</b> . . . . .	<b>4</b>
1, rész . . . . .	4
2, rész . . . . .	4
3, rész . . . . .	5

## Part I

# IPC mechanizmus - Szignálkezelés

## 1. feladat

A tanult rendszerhívásokkal (`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_RDWR`
- 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.

## 2. feladat

Készítse el a következő feladatot, melyben egy szignálkezelő több szignált is tud kezelni:

### A, rész

Készítsen egy szignál kezelőt (handleSignals), amely a SIGINT vagy SIGQUIT jelek fogására vagy kezelésére képes.

### B, rész

Ha a felhasználó SIGQUIT jelet generál (akár kill paranccsal, akár billentyűzetről. A kezelő egyszerűen kiírja az üzenetet visszatérési értékét – a konzolra.

### C, rész

Ha a felhasználó először generálja a SIGINT jelet (akár kill paranccsal, akár billentyűzetről a CTRL + C), akkor a jelet úgy módosítja, hogy a következő alkalommal alapértelmezett műveletet hajtson végre (a SIG-DFL) – kiírás a konzolra.

### D, rész

Ha a felhasználó másodszor generálja a SIGINT jelet, akkor végrehajt egy alapértelmezett műveletet, amely a program befejezése - kiírás a konzolra.

# 1. rész

[illegible][illegible]

4 ms	P1	P2	P3	P4	
Érkezés	0, 4, 15	0	2, 11	5	P2, P3, P1
CPU idő	24, 20, 16	3	6, 2	3	P3, P1, P4
Indulás	0, 11, 20	4	7, 18	15	P1, P4, P3
Befejezés	4, 15, 36	7	11, 20	18	P4, P3, P1
Várakozás	0, 7, 5	4	5, 7	10	P3, P1
	4, 11, 21	7	9, 9	13	
<b>Algoritmus neve: RR</b>					
CPU kihasználtság	156,4				
Konvergenzálási időköz átlaga	10				
Várakozási időköz átlaga	7				
Válaszidőköz átlaga	95,6				

RR megoldás

## 2. rész

Mentés. neptunkod-gyak9-1.c

### 3, rész

Írjon C nyelvű programot, amelyik a SIGTERM-hez hozzárendel egy fv-t., amelyik kiírja az int paraméter értéket, majd végtelen ciklusban fusson, 3 sec-ig állandóan blokkolódva elindítás után egy másik shell-ben kill paranccsal (SIGTERM) próbálja terminálni, majd SIGKILL-el.”

Mentés. neptunkod-gyak9-2.c