

1.feladat

Round Robin nélkül

| | 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 |
| Starting point | 60 | 0 | 60 | 0 | 60 | 0 | 60 | 0 | | |
| 1 | 60 | 1 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| 2 | 60 | 2 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| 3 | 60 | 3 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| | | | | | | | | | | |
| 10 | 60 | 10 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| 11 | 60 | 11 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| | | | | | | | | | | |
| 99 | 60 | 99 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| 100 | 72 | 50 | 60 | 0 | 60 | 0 | 60 | 0 | B | B |
| 101 | 72 | 50 | 60 | 1 | 60 | 0 | 60 | 0 | B | B |
| | | | | | | | | | | |
| 110 | 72 | 50 | 60 | 10 | 60 | 0 | 60 | 0 | B | B |
| 111 | 72 | 50 | 60 | 11 | 60 | 0 | 60 | 0 | B | B |
| | | | | | | | | | | |
| 199 | 72 | 50 | 60 | 99 | 60 | 0 | 60 | 0 | B | B |
| 200 | 66 | 25 | 72 | 50 | 60 | 0 | 60 | 0 | C | C |
| 201 | 66 | 25 | 72 | 50 | 60 | 1 | 60 | 0 | C | C |

Round Robin

| | 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 |
| Starting point | 60 | 0 | 60 | 0 | 60 | 0 | 60 | 0 | | |
| 1 | 60 | 1 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| 2 | 60 | 2 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| 3 | 60 | 3 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| | | | | | | | | | | |
| 9 | 60 | 9 | 60 | 0 | 60 | 0 | 60 | 0 | A | A |
| 10 | 60 | 10 | 60 | 0 | 60 | 0 | 60 | 0 | A | B |
| 11 | 60 | 10 | 60 | 1 | 60 | 0 | 60 | 0 | B | B |
| | | | | | | | | | | |
| 20 | 60 | 10 | 60 | 10 | 60 | 0 | 60 | 0 | B | C |
| 21 | 60 | 10 | 60 | 10 | 60 | 1 | 60 | 0 | C | C |
| | | | | | | | | | | |
| 30 | 60 | 10 | 60 | 10 | 60 | 10 | 60 | 0 | C | D |
| 31 | 60 | 10 | 60 | 10 | 60 | 10 | 60 | 1 | D | D |
| | | | | | | | | | | |
| 40 | 60 | 10 | 60 | 10 | 60 | 10 | 60 | 10 | D | A |
| 41 | 60 | 11 | 60 | 10 | 60 | 10 | 60 | 10 | A | A |
| | | | | | | | | | | |
| 50 | 60 | 20 | 60 | 10 | 60 | 10 | 60 | 10 | A | B |
| 51 | 60 | 20 | 60 | 11 | 60 | 10 | 60 | 10 | B | B |
| | | | | | | | | | | |
| 60 | 60 | 20 | 60 | 20 | 60 | 10 | 60 | 10 | B | C |
| 61 | 60 | 20 | 60 | 20 | 60 | 11 | 60 | 10 | C | C |
| | | | | | | | | | | |
| 70 | 60 | 20 | 60 | 20 | 60 | 20 | 60 | 10 | C | D |
| 71 | 60 | 20 | 60 | 20 | 60 | 20 | 60 | 11 | D | D |
| | | | | | | | | | | |
| 80 | 60 | 20 | 60 | 20 | 60 | 20 | 60 | 20 | D | A |
| 81 | 60 | 21 | 60 | 20 | 60 | 20 | 60 | 20 | A | A |
| | | | | | | | | | | |
| 90 | 60 | 30 | 60 | 20 | 60 | 20 | 60 | 20 | A | B |
| 91 | 60 | 30 | 60 | 21 | 60 | 20 | 60 | 20 | B | B |
| | | | | | | | | | | |
| 99 | 60 | 30 | 60 | 29 | 60 | 20 | 60 | 20 | B | C |
| 100 | 68 | 15 | 68 | 15 | 65 | 10 | 75 | 10 | C | C |
| 101 | 68 | 15 | 68 | 15 | 65 | 11 | 75 | 10 | C | C |
| | | | | | | | | | | |
| 190 | 68 | 15 | 68 | 15 | 65 | 100 | 75 | 10 | C | C |
| 191 | 68 | 15 | 68 | 15 | 65 | 101 | 75 | 10 | C | C |
| | | | | | | | | | | |
| 199 | 68 | 15 | 68 | 15 | 65 | 109 | 75 | 10 | C | C |
| 200 | 72 | 7 | 72 | 7 | 92 | 55 | 88 | 5 | C | A |
| 201 | 72 | 8 | 72 | 7 | 92 | 55 | 88 | 5 | A | A |

2.feladat

A tanult rendszerhívásokkal (open(), read()/write(), close()) – ezek fogják a rendszerhívásokat tovább hívni) írjanak egy neptunkod_openclose.c programot, amely megnyit egy fájlt – neptunkod.txt (O_RDWR megnyitási móddal), tartalma: hallgató neve, szak , neptunkod.

G2SKZ4_openclose.c program:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>

int main(void) {
    int fd;
    char *buf[100];
    int len;

    if ((fd = open("G2SKZ4.txt", O_RDWR)) < 0) {
        perror("open hiba");
        return -1;
    }

    if ((len = read(fd, buf, sizeof(buf))) != 0) {
        if (len == -1 && errno != EINTR) {
            perror("read hiba");
            return -1;
        }
        printf("Beolvasva: %d byte\nTartalma: %s\n", len, buf);
    }

    if (lseek(fd, 0, SEEK_SET) < 0) {
        perror("lseek hiba");
        return -1;
    }
    printf("Pozicio a fajl elejere mozgatva\n");

    if ((len = write(fd, "Hello World", 11)) != 11) {
        perror("write hiba");
        return -1;
    }
    printf("%d byte kiírva\n", len);

    close(fd);
    return 0;
}
```