

1,

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

```
2,  
//  
// main.c  
// H22TQ1_openclose  
//  
// Created by Balogh Bianka on 2021. 03. 09..  
//
```

```
#include <stdio.h>  
#include <stdlib.h>
```

```
#include <sys/types.h>  
#include <sys/stat.h>  
#include <fcntl.h>
```

```
#include <unistd.h>
```

```
#define FAJL "H22TQ1.txt"
```

```
int main(void)  
{  
    int fileHandle = open(FAJL, O_RDWR);  
    if(fileHandle == -1)  
    {  
        printf("Nem sikerult megnyitni a(z) \"%s\" fajlt!\n", FAJL);  
        return 1;  
    } else printf("Megnyitottam a(z) \"%s\" fajlt!\n", FAJL);  
  
    char tartalom[64];  
    int olvasott = read(fileHandle, tartalom, sizeof(tartalom));  
  
    tartalom[olvasott] = '\0';  
    printf("beolvasott tartalom: \"%s\" osszesen \"%i\" byte.\n", tartalom, olvasott);  
  
    lseek(fileHandle, 0, SEEK_SET);  
    printf("A fajl elejere allitottuk a mutatot\n");  
  
    char szoveg[] = "Rendszerhivassal iras fajlba";  
    int irt = write(fileHandle, szoveg, sizeof(szoveg));  
    printf("A fajlba irtuk a(z) \"%s\" szoveget. osszesen \"%i\" byte.\n", szoveg, irt);  
    close(fileHandle);  
    return 0;  
  
}
```

