

JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: Dobra Gábor
Neptunkód: XQBTIW

1. Adott az alábbi terhelés esetén a rendszer. Határozza meg az *indulás*, *befejezés*, *várakozás/átlagos várakozás és körülfordulás/átlagos körülfordulás*, *válasz/átlagos válaszidő* és a *CPU kihasználtság* értékeket az FCFS ütemezési algoritmusok mellett! (cs: 0,1ms; sch: 0,1ms)

	P1	P2	P3	P4
Érkezés	0	8	12	20
CPU idő	15	7	26	10
Indulás				
Befejezés				
Várakozás				

Ábrázolja Gantt diagram segítségével az *aktív/várakozó processzek* futásának menetét.

1. Feladat: OS algoritmusok

FCFS (First Came First Served):

A legrégebben várakozó feladatot választja ki végrehajtásra. Nem preemptív algoritmus. A várakozási sor azonos végéről veszi ki mindig a feladatokat.

Az ütemezési algoritmus lefutása:

	Érkezés	Hossz	Indulás	Befejezés	Várakozás	Körülfordulás	Válaszidő	CPU %
P1	0	15	0	15	0	15	0	
P2	8	7	15	22	7	14	7	
P3	12	26	22	48	10	36	10	
P4	20	10	48	58	28	38	28	
Átlagok					11,25	25,75	11,25	99,3%

Az ütemezéssel kapcsolatos metrikák értéke:

Metrika	Érték
CPU kihasználtság ²	4 context switch, $(58,4 - 0,4) / 58,4 = 99,3 \%$
Körülfordulási idők átlaga	$(15+14+36+38) / 4 = 25,75 \text{ ms}$
Várakozási idők átlaga	$(0+7+10+28) / 4 = 11,25 \text{ ms}$
Válaszidők átlaga	$(0+7+10+28) / 4 = 11,25 \text{ ms}$

CPU kihasználtság:

A hasznos munkával töltött idő aránya az összes időhöz képest.

Várakozási idő:

Az összes idő, amit a feladat várakozással töltött. A munka jellegéből és a végrehajtás részleteiből adódóan statisztikai ingadozás lehet, ezért átlagos várakozási időt számolunk leggyakrabban.

$${}^t\text{várakozás} = {}^t\text{futásra kész} + {}^t\text{más, nem futó állapotok}^{[s]}$$

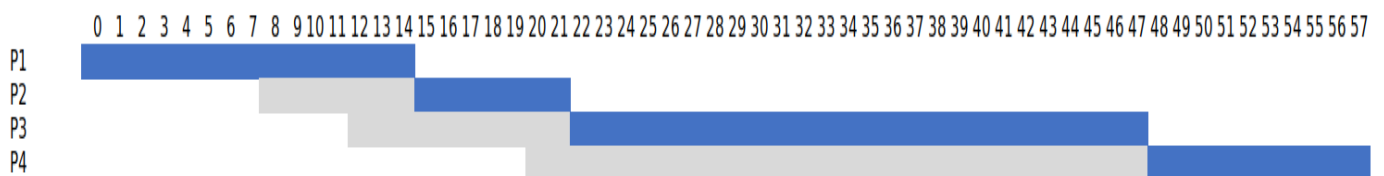
Körülfordulási idő:

Egy feladatra vonatkozóan a rendszerbe helyezéstől a teljesítésig eltelt idő. A munka jellegéből és a végrehajtás részleteiből adódóan statisztikai ingadozás lehet, ezért átlagos körülfordulási időt számolunk leggyakrabban.

$${}^t\text{körülfordulás} = {}^t\text{CPU, végrehajtás} + {}^t\text{várakozás}^{[s]}$$

Válaszidő:

A feladat megkezdésétől az első kimenetek produkálásáig eltelt idő. A munka jellegéből és a végrehajtás részleteiből adódóan statisztikai ingadozás lehet, ezért átlagos válaszidőt számolunk leggyakrabban.

Gantt diagram:

2. feladat: IPC mechanizmus

Írjon C nyelvű programokat, ami

SIGUSR1 signal érkeztekor növeli egy változó értékét

SIGUSR2 signal hatására pedig kiírja a képernyőre az aktuális értéket, a másik program pedig futtatáskor signalt küld az adott processznek.

```
1 #include<stdio.h>
2 #include<signal.h>
3 #include <stdlib.h>
4
5 int counter = 0;
6 void handle_signals(int sig) {
7     if (sig == SIGUSR1) {
8         counter += 1;
9         printf("SIGUSR1 jel elkapva, valtozo noveleve.\n");
10    } else if (sig == SIGUSR2) {
11        printf("SIGUSR2 jel elkapva, valtozo erteke: %d\n", counter);
12    }
13 }
14
15 int main()
16 {
17     signal(SIGUSR1, handle_signals);
18     signal(SIGUSR2, handle_signals);
19
20     while (1) ;
21     return 0;
22 }
```

```
1 #include<stdio.h>
2 #include<signal.h>
3 #include <stdlib.h>
4
5
6 int main()
7 {
8     pid_t pid;
9     int signal;
10    printf("Ml a folyamat PID-je?\n");
11    scanf("%d", &pid);
12
13    while(1) {
14        printf("Milyen jelet küldjünk? 1 - SIGUSR1 2 - SIGUSR2 0 - kilépés\n");
15        scanf("%d", &signal);
16
17        if (signal == 0) {
18            printf("Program befejezése.\n");
19            break;
20        } else if (signal == 1) {
21            kill(pid, SIGUSR1);
22        } else if (signal == 2) {
23            kill(pid, SIGUSR2);
24        }
25
26        printf("Signal elküldve.\n");
27    }
28
29
30    return 0;
31 }
```

A program a következő képpen működik:

Van egy *reciever* program ami a háttérben fut, ennek meg kell néznünk a program PID-ét.

Main	I/O										
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command^
901	hp	20	0	101M	700	0	S	0.0	0.0	0:00.00	(sd-pam)
293550	hp	20	0	2364	572	504	R	103	0.0	0:38.07	./receiver
900	hp	20	0	21952	4940	3256	S	0.0	0.1	0:10.31	/lib/systemd/systemd --user
239	root	19	-1	92836	14464	13400	S	0.0	0.4	2:02.28	/lib/systemd/systemd-journald
778	root	20	0	16788	2760	2052	S	0.0	0.1	2:09.35	/lib/systemd/systemd-logind
677	systemd-re	20	0	24264	3168	1788	S	0.0	0.1	2:19.65	/lib/systemd/systemd-resolved
678	systemd-ti	20	0	90220	524	4	S	0.0	0.0	0:02.25	/lib/systemd/systemd-timesyncd
697	systemd-ti	20	0	90220	524	4	S	0.0	0.0	0:00.65	/lib/systemd/systemd-timesyncd
273	root	20	0	24372	1592	916	S	0.0	0.0	0:21.45	/lib/systemd/systemd-udev
281508	hp	20	0	16.8G	158M	91872	S	0.0	4.1	4:51.02	/opt/google/chrome/chrome --enable-
281523	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281531	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281532	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.29	/opt/google/chrome/chrome --enable-
281533	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.32	/opt/google/chrome/chrome --enable-
281537	hp	20	0	16.8G	158M	91872	S	0.0	4.1	1:10.28	/opt/google/chrome/chrome --enable-
281538	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:01.00	/opt/google/chrome/chrome --enable-
281539	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281540	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.01	/opt/google/chrome/chrome --enable-
281542	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.12	/opt/google/chrome/chrome --enable-
281543	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281544	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281545	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281548	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281551	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.19	/opt/google/chrome/chrome --enable-
281552	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:08.29	/opt/google/chrome/chrome --enable-
281553	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
281567	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:01.15	/opt/google/chrome/chrome --enable-
281597	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.36	/opt/google/chrome/chrome --enable-
281598	hp	20	0	16.8G	158M	91872	S	0.0	4.1	0:00.00	/opt/google/chrome/chrome --enable-
F1Help	F2Setup	F3Search	F4Filter	F5Tree	F6SortBy	F7Nice	-F8Nice	+F9Kill	F10Quit		

Ezután elindítjuk a *sender* programot, ami bekéri a program PID-ét, utána az 1-et vagy a 2-őt.

```
receiver receiver1.c sender sender.c
hp@hp-HP-Pavilion-Sleekbook-15:~/Asztal/01$ gcc -o sender sender.c
hp@hp-HP-Pavilion-Sleekbook-15:~/Asztal/01$ ./sender
Mi a folyamat PID-je?
293550
Milyen jelet küldjünk? 1 - SIGUSR1 2 - SIGUSR2 0 - kilépés
```

Ha 1-et írunk, akkor az növel a *recevier* programban egy számlálót, ha 2-őt akkor pedig kiírja a változó értéket, 0-val kilépünk a programból.

```
receiver receiver1.c sender sender.c
hp@hp-HP-Pavilion-Sleekbook-15:~/Asztal/01$ gcc -o receiver receiver1.c
hp@hp-HP-Pavilion-Sleekbook-15:~/Asztal/01$ ./receiver
SIGUSR1 jel elkapva, valtozo noveleve.
SIGUSR2 jel elkapva, valtozo erteke: 1
SIGUSR1 jel elkapva, valtozo noveleve.
SIGUSR1 jel elkapva, valtozo noveleve.
SIGUSR2 jel elkapva, valtozo erteke: 3
SIGUSR2 jel elkapva, valtozo erteke: 3
□
```