

OPERÁCIÓS RENDSZEREK

Processzorütemezés

4. gyakorlat

Processzorütemezési algoritmusok

1. feladat: A rendszerben öt folyamat van, minden egyik a 0 időpontban érkezik.

	Érkezési idő	CPU-löketidő (ms)
P1	0	10
P2	0	29
P3	0	3
P4	0	7
P5	0	12

- Milyen a folyamatok végrehajtási sorrendje, ha a rendszer **FCFS**, **SJF**, **SRTF** és **RR** (időszak = 10 ms) ütemezési algoritmust használ? Rajzolja fel a megfelelő Gantt-diagramokat!
- Melyik algoritmus adja a legkisebb átlagos várakozási időt?
- Az **FCFS** algoritmusnál határozza meg az átlagos körülfordulási időt!

Processzorütemezési algoritmusok

2. feladat: A rendszerben három folyamat van, az érkezési időket, valamint a CPU-löketidőket az alábbi táblázat tartalmazza:

	Érkezési idő	CPU-löketidő (ms)
P1	0.0	8
P2	0.4	4
P3	1.0	1

- Milyen a folyamatok végrehajtási sorrendje, ha a rendszer **FCFS** és **SJF** ütemezési algoritmust használ? Rajzolja fel a megfelelő Gantt-diagramokat!
- Melyik algoritmus adja a kisebb átlagos körülfordulási időt?
- Számítsa ki az átlagos körülfordulási időt, ha a CPU először 1 ms ideig tétlen marad, majd ezután a rendszer **SJF** ütemezést használ.
Megjegyzés: Ne feledje, hogy a **P1** és **P2** folyamatok ebben az üresjáratban várakoznak, így várakozási idejük megnőhet!

Processzorütemezési algoritmusok

3. feladat: A rendszerben öt folyamat van, az érkezési időket, a CPU-löketidőket és a prioritásokat az alábbi táblázat tartalmazza.

	Érkezési idő	CPU-löketidő (ms)	Prioritás
P1	0	2	2
P2	0	1	1
P3	0	8	4
P4	0	4	2
P5	0	5	3

- Ábrázolja a folyamatok végrehajtási sorrendjét Gantt-diagramokkal, ha a rendszer **FCFS**, **SJF**, **nem megszakítható prioritásos** és **RR** (időszelet = 2 ms) ütemezési algoritmust használ!
- Az egyes algoritmusoknál mennyi a folyamatok *körülfordulási ideje*?
- Az egyes algoritmusoknál mennyi a folyamatok *várakozási ideje*?
- Melyik algoritmus adja a legkisebb *átlagos várakozási időt*?

Processzorütemezési algoritmusok

4. feladat: A rendszerben hat folyamat van, és megszakítható, RR (időszelet: 10 ms) ütemezési algoritmust használ. Az érkezési időket, a CPU-löketidőket és a prioritásokat az alábbi táblázat tartalmazza.

	Érkezési idő	CPU-löketidő (ms)	Prioritás
P1	0	20	40
P2	25	25	30
P3	30	25	30
P4	60	15	35
P5	100	10	5
P6	105	10	10

E folyamatokon kívül a rendszernek van egy üresjárati feladata is (amely nem fogyaszt CPU-erőforrást, és P_{idle} -ként van azonosítva). Ennek a feladatnak a prioritása 0, és minden olyan esetben ütemezhető, amikor a rendszernek nincs más futtatható folyamata. Ha egy folyamatot megelőz egy magasabb prioritású folyamat, akkor a megelőző folyamat a sor végére kerül.

Processzorütemezési algoritmusok

4. feladat: A rendszerben hat folyamat van, és megszakítható, RR (időszelet: 10 ms) ütemezési algoritmust használ. Az érkezési időket, a CPU-löketidőket és a prioritásokat az alábbi táblázat tartalmazza.

	Érkezési idő	CPU-löketidő (ms)	Prioritás
P1	0	20	40
P2	25	25	30
P3	30	25	30
P4	60	15	35
P5	100	10	5
P6	105	10	10

- a) Ábrázolja a folyamatok végrehajtási sorrendjét Gantt-diagrammal!
- b) Határozza meg a folyamatok *körülfordulási idejét*!
- c) Határozza meg a folyamatok *várakozási idejét*!
- d) Hány százalékos a CPU kihasználtsága?

Processzorütemezési algoritmusok

5. feladat: A rendszerben öt folyamat van, mindegyik a 0 időpontban érkezik.

	Érkezési idő	CPU-löketidő (ms)
P1	0	9
P2	0	6
P3	0	3
P4	0	5
P5	0	X

Milyen legyen a folyamatok végrehajtási sorrendje, ha azt akarjuk, hogy az átlagos válaszidő a lehető legkisebb legyen?

Processzorütemezési algoritmusok

6. feladat: A kötegelt rendszerben öt folyamat van, az érkezési időket, a futási időket és a prioritásokat az alábbi táblázat tartalmazza.

	Érkezési idő	Futási idő (min)	Prioritás
A	0	10	3
B	0	6	5
C	0	2	2
D	0	4	1
E	0	8	4

Adja meg az átlagos körülfordulási időket, ha a rendszer **FCFS**, **SJF**, **nem megszakítható prioritásos** és **RR** (tegyük fel, hogy a rendszer multiprogramozott, és minden feladat megkapja a megfelelő részét a CPU-ból) ütemezési algoritmust használ!

Processzorütemezési algoritmusok

7. feladat: Egy lágy valós idejű rendszernek (**soft real-time system**) négy periodikus eseménye van, egyenként 50, 100, 200 és 250 ms periódussal. Tegyük fel, hogy a négy esemény rendre 35, 20, 10 és X ms CPU-időt igényel. Határozza meg az X legnagyobb értékét, amelyre a rendszer még ütemezhető!