

1) Nepreemptivní SJF algoritmus:

P1 vznik v case 0, trvani 7

P2 vznik v case 2, trvani 4

P3 vznik v case 4, trvani 1

P4 vznik v case 5, trvani 4

Který proces pobezi v case 10? -> P2

t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇	t ₈	t ₉	t ₁₀
P1	P1	P1	P1	P1	P1	P1	P3	P2	P2

2) RAG s cyklem (P1->R1->P2->R2->P3->R3->P1) - dojde k uvážnutí

3) IPC

správně: semafor, sdílená paměť i roura

špatně: registry

4) mikrojádru:

Ze slidů (přednáška 4, slidy 19 a 20):

Microkernel System Structure

Malé jádro OS plnící pouze několik málo nezbytných funkcí

- primitivní správa paměti (adresový prostor)
- komunikace mezi procesy – Interprocess communication (IPC)

Většina funkcí z jádra se přesouvá do „uživatelské“ oblasti

- ovladače HW zařízení, služby systému souborů, virtualizace paměti ...
- mezi uživatelskými procesy se komunikuje předáváním zpráv

Výhody mikrojádra

- snadná přenositelnost OS, jádro je malé
- vyšší spolehlivost (moduly mají jasné API a jsou snadněji testovatelné)
- vyšší bezpečnost (méně kódu OS běží v režimu jádra)
- flexibilita (jednodušší modifikace, přidání, odebrání modulů)
- všechny služby jsou poskytovány jednotně (výměnou zpráv)

Nevýhoda mikrojádra

- zvýšená režie
- volání služeb je nahrazeno výměnou zpráv mezi procesy

5) page table (správně 0/4/v 1/x/i 2/6/v 3/x/i 4/x/i 5/9/v 6/x/i 7/x/i)

(přednáška 11, slide 13)

6) Proces typicky prechazi ze stavu bezici (running) do pripravený (ready) pri:

správně: pri plánování pomocou algoritmu Round Robin RR

nesprávně: Synchronni I/O, Asynchrone I/O...

7) Digitální podpis ovladačů jádra:

-> uplatňuje se pro zvýšení bezpečnosti

-> je ve Windows7

Špatný možnosti: něco s DOSem a že už se nepoužívá, bývá něčím nahrazen...

8) Paměť pro vyrovnávání bloků vstup/výstup

buffer (dobře)

dál bylo:

cache

Best-fit

Wait - For cyklus

9) Predávanie argumentov pri systémových volaniach je realizované

dobře: registre, zásobník, pointer na strukturu s datami uloženými v paměti patriacej procesu

zle: IN/OUT instrukcí, kritické sekce

(přednáška 3, slide 24)

10)

Plánovací rozhodnutí může vydat v okamžiku, kdy proces:

1. přechází ze stavu běžící do stavu čekající
2. přechází ze stavu běžící do stavu připravený
3. přechází ze stavu čekající do stavu připravený
4. končí

Případy 1 a 4 se označují jako nepreemptivní plánování (plánování bez předbíhání)

Případy 2 a 3 se označují jako preemptivní plánování (plánování s předbíháním) - na tohle se ptal

(přednáška 7, slide 6)