

Operacijski sistemi



Razvrščanje

Razvrščanje

- Razvrščanje

- odločanje o tem, kateri proces se razvrsti na viru
 - npr. pridobi procesor, pomnilnik, napravo ipd.
- ključno za zagotavljanje *večopravnosti*

- Pogledi na razvrščanje

obdobje	pomen	angleško
dolgoročno	razvrščanje poslov	job scheduling
srednjeročno	menjavanje procesov	swapping
kratkorочно	razvrščanje procesov na procesorju	CPU scheduling

Razvrščanje

- Razvrščanje poslov (dolgoročno)
 - paketna obdelava
 - enota obdelave je **posel** (opravilo za obdelavo)
 - posli čakajo (na disku), da bodo izbrani za obdelavo
 - ob izbiri se naložijo in začnejo izvajati
 - razvrščevalnik poslov (job scheduler)
 - del OS, ki skrbi za razvrščanje poslov
 - navadno ni podprto na modernih OS
 - analogija z ročnim zagonom programa

Razvrščanje

- **Menjavanje procesov (srednjeročno)**
 - uporaba diska za umikanje procesov
 - če pomnilnika primanjkuje, proces lahko umaknemo na disk in s tem sprosimo njegove vire
 - procese lahko tudi naložimo nazaj z diska v pomnilnik
 - menjalnik (swapper)
 - izbira procese za umik in za vračanje
 - skrb za enakomerno obremenitev virov

Razvrščanje

- Razvrščanje na procesorju (kratkoročno)
 - odločanje o tem, kateri proces dobi možnost izvajanja na procesorju
 - ključno za zagotavljanje večopravnosti
 - razvrščevalnik (širše)
 - del upravitelja procesov
 - razvrščevalnik (ožje, scheduler)
 - izbira enega izmed pripravljenih procesov
 - dodeljevalnik (dispatcher)
 - preklop procesa iz trenutnega na izbranega

Razvrščanje

- **Kdaj se aktivira razvrščevalnik?**

- končanje procesa
- proces je onemogočen (postane čakajoč)
- potek časovne rezine
- konec čakanja
 - proženje čakajočega dogodka
 - vstop v vrsto pripravljenih procesov
- stvaritev novega procesa
 - vstop v vrsto pripravljenih procesov

Razvrščanje

- Časovna rezina

- **računsko intenzivna opravila** (CPU bound tasks)

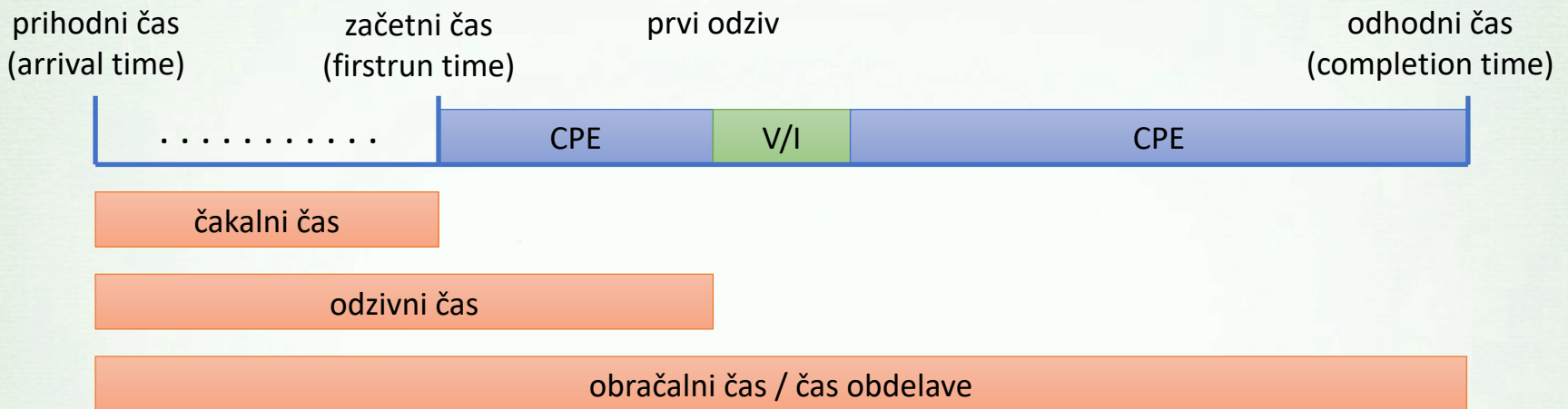
- preferirajo daljše časovne rezine
 - s tem omejijo stroške preklopa procesa
 - izkoriščenost procesorja je višja

- **interaktivna opravila** (I/O bound tasks)

- preferirajo krajše časovne rezine
 - procesi prej pridejo na vrsto
 - boljša uporabniška izkušnja

Mere zmogljivosti

- Paketni sistemi



Čakalni čas (waiting time)

$$T_{\text{čakalni}} = T_{\text{začetni}} - T_{\text{prihodni}}$$

Odzivni čas (response time)

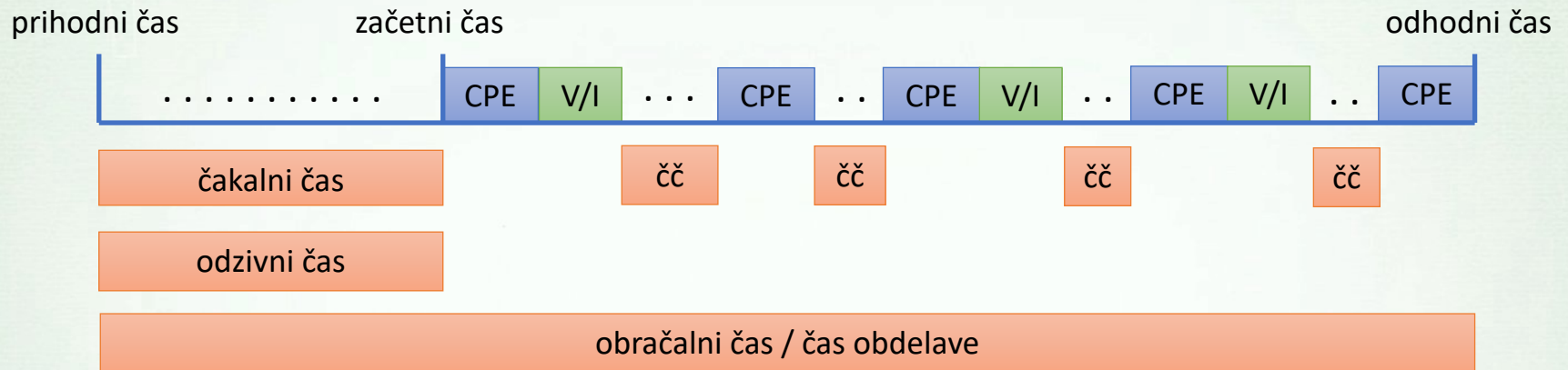
$$T_{\text{odzivni}} = T_{v/i} - T_{\text{prihodni}}$$

Čas obdelave (turnaround time)

$$T_{\text{obdelave}} = T_{\text{končni}} - T_{\text{prihodni}}$$

Mere zmogljivosti

- Interaktivni sistemi – časovno dodeljevanje



Čakalni čas

celoten čas prebit v stanju pripravljen

Odzivni čas

predpostavimo: $T_{začetni} \simeq T_{v/i}$

$T_{odzivni} = T_{začetni} - T_{prihodni}$

Čas obdelave (turnaround time)

$T_{obdelave} = T_{končni} - T_{prihodni}$

Mere zmogljivosti

- Ostale mere
 - **izkoriščenosti procesorja** (processor utilization)
 - delež zaposlenosti procesorja
 - **prepustnost sistema** (system throughput)
 - število obdelanih procesov v danem časovnem obdobju
 - **poštenost** (fairness)
 - enakomernost delitve procesorja procesom glede na prioriteto procesa

Razvrščevalni algoritmi

- Osnovni algoritmi
 - FCFS – prvi pride prvi melje
 - SJF – najkrajši posel najprej
 - PSJF – prekinjevalni najkrajši posel najprej
 - RR – krožno razvrščanje
- Prednostni algoritmi
 - HPF – najvišja prioriteta najprej
 - razvrščanje s prepustnicami
 - koračno razvrščanje
- Praktični algoritmi
 - MLFQ – večnivojska odzivna vrsta
 - razvrščevalnik Linux $O(1)$
 - CFS – popolnoma pošteno razvrščanje

Osnovni algoritmi

FCFS – first come, first served
FIFO – first in, first out

- FCFS – prvi pride, prvi melje
 - **proces, ki je prej pripravljen, prej dobi procesor**
 - pripravljeni procesi vstopajo na konec vrste
 - za izvajanje se vzame proces iz čela vrste
 - lastnosti
 - razvrščanje brez odvzemanja
 - odzivni čas: **slab**
 - čas obdelave: **dober**
 - efekt „konvoja“

Osnovni algoritmi

SJF – shortest job first

- SJF – najkrajši posel najprej
 - **najkrajši pripravljeni posel prej dobi procesor**
 - intuicija: procesi se hitreje *obračajo*, naprave prej dobijo zaposlitev
 - potrebno je vnaprejšnje poznavanje dolžine poslov
 - lastnosti
 - razvrščanje brez odvzemanja
 - algoritem je celo optimalen v tem primeru
 - odzivni čas: **slab**
 - čas obdelave: **odličen**

Osnovni algoritmi

PSJF – preemptive shortest job first
SRTF – shortest remaining-time first
STCF – shortest time-to-completion first

- **PSJF – prekinjevalni najkrajši posel najprej**
 - ali „kdor se prej konča, najprej“
 - **pripravljeni posel z najkrajšim preostankom časa prej dobi procesor**
 - potrebno vnaprejšnje poznavanje dolžine poslov
 - lastnosti
 - razvrščanje z odvzemanjem
 - **odzivni čas: slab**
 - **obračalni čas: odličen**

Osnovni algoritmi

RR – round robin

- **RR – krožno razvrščanje**
 - **pripravljene posle krožno razvrščamo zaporedoma, vsakega za nekaj časa**
 - pripravljeni procesi vstopajo na konec vrste
 - za izvajanje se vzame proces iz čela vrste
 - proces se izvaja le za določen čas, nato se prekine
 - časovna rezina
 - dovolj kratka, da zadosti odzivnosti
 - dovolj dolga, da opraviči čas preklopa
 - lastnosti
 - razvrščanje z odvzemanjem
 - odzivni čas: **odličen**
 - čas obdelave: **slab**

Osnovni algoritmi

- Ocena časa izvajanja

- ideja

- naslednje trajanje je podobno predhodnemu

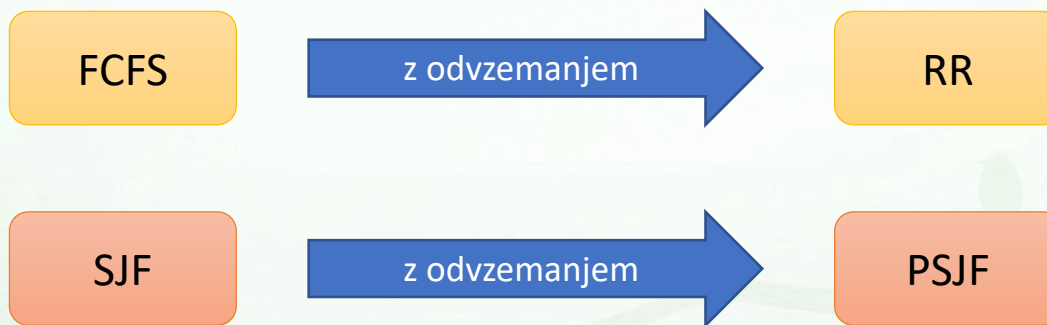
- izvedba

- t ... trajanje zadnjega teka procesa
 - t' ... ocena trajanja teka (eksponentno povprečje)
 - α ... faktor pozabljanja
 - $\alpha = 1$: preteklost se ne upošteva
 - $\alpha = 0$: trajanje zadnjega teka nima vpliva
 - $t' := \alpha \cdot t + (1 - \alpha) \cdot t'$

Osnovni algoritmi

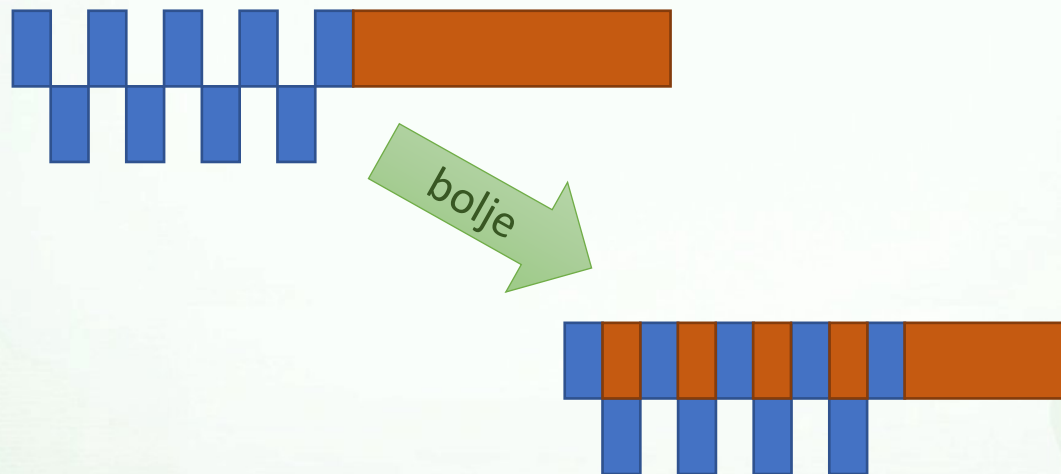
- Primerjava algoritmov

Algoritem	Odzivni čas	Čas obdelave
FIFO	slab	dober
SJF	slab	odličen
PSJF	slab	odličen
RR	odličen	slab



Osnovni algoritmi

- Vključevanje V/I operacij
 - tekom V/I operacij je proces blokiran
 - procesor je neizkoriščen
 - prekrivanje izvajanja



Prednostni algoritmi

- Prednostno razvrščanje
 - razvrščevalni algoritmi, ki dajejo prednost izbranim procesom glede na neko lastnost
- Upoštevanje prioritete procesov
 - **notranja prioriteta**
 - odvisna od lastnosti procesa
 - npr. trajanje procesa (SJF), preostanek teka (PSJF), velikost procesa, št. odprtih datotek ipd.
 - **zunanja prioriteta**
 - določena s strani uporabnika
 - npr. ukaza `nice` in `renice` za nastavljanje prioritete v Unix sistemih

Prednostni algoritmi

HPF – highest priority first

- **HPF – najvišja prioriteta najprej**
 - **pripravljeni posel z najvišjo prioriteto dobi procesor najprej**
 - **razvrščevalnik brez ali z odvzemanjem**
 - vstop procesa v vrsto pripravljenih
 - vstopni proces z višjo prioriteto kot izvajajoči se

razvrščanje	aktivacija razvrščevalnika	komentar
sodelovalno / brez odvzemanja	ne	počakamo, da trenutni proces prepusti procesor
prekinjevalno / z odvzemanjem	da	če ima prispeli proces višjo prioriteto, potem zamenja trenutnega

Prednostni algoritmi

- Upoštevanje prioritete procesov
 - težava: **stradanje**
 - procesi z višjo prioriteto nenehno **prehitevajo** procese z nižjo prioriteto
 - nek proces ne pride nikoli na vrsto
 - rešitev: **staranje**
 - procesom, ki ne dobijo procesorja, vsake toliko časa povečamo prioriteto

Prednostni algoritmi

SIRO – service in random order

- Naključno razvrščanje
 - **razvrščanje po neki shemi naključnosti**
 - možno odvzemanje ali pa tudi ne
 - pravičnost razvrščanja
 - dolgoročno pravično glede *prioriteto*

Prednostni algoritmi

lottery scheduling

- Razvrščanje s prepustnicami
 - **določanje procesa z loterijo**
 - stvaritev procesa
 - proces i ima (dobi ob stvaritvi) p_i prepustnic
 - procesi z višjo prioriteto dobijo več prepustnic
 - razvrščanje
 - razvrščevalnik naključno izbere eno prepustnico in procesu lastniku prepustnice dodeli procesor
 - procesi lahko prepuščajo prepustnice drugim
 - npr. odjemalec prepusti prepustnice strežniku
 - verjetnost izbire k -tega procesa $\frac{p_k}{\sum_{i=1}^n p_i}$

Prednostni algoritmi

stride scheduling

- Koračno razvrščanje

- proces, ki je najmanj *prehodil*, najprej
- stvaritev procesa
 - procesom dodelimo *dolžino koraka*
 - procesi z višjo prioriteto imajo krajši korak
- razvrščanje
 - procesu, ki prehodil najmanj, se dodeli procesor
 - koraki se seštevajo v skupno pot
- izziv
 - kaj narediti, ko pride nov proces

Praktični algoritmi

MLQ – multilevel queue

- MLQ – večnivojska vrsta

- skupine opravil: interaktivna in paketna opravila
 - različni razvrščevalniki so različno primerni
- večnivojska vrsta oz. več vrst
 - vrste padajoče urejene po prioriteti
 - naraščajoče urejene po časovni rezini
 - znotraj posamezne vrste se uporablja RR

		pri.	č. r.
→	sistemski procesi	3	10 ms
→	interaktivni procesi	2	20 ms
→	paketni procesi	1	40 ms
→	servisi	0	80 ms

Praktični algoritmi

MLFQ – multi-level feedback queue

- MLFQ – večnivojska odzivna vrsta
 - MLF + prehajanje poslov med vrstami
 - prehajanje med vrstami
 - poslu ob prihodu dodelimo najvišjo prioriteto
 - poslu, ki porabi celotno časovno rezino, dekrementiramo prioriteto
 - posel, ki ne porabi celotne časovne rezine, ostane na isti prioriteti
 - po določenem času, vse posle prestavimo na najvišjo prioriteto
 - preprečevanje stradanja
 - reset naučenega stanja
- optimizira odzivni čas in čas obdelave

Praktični algoritmi

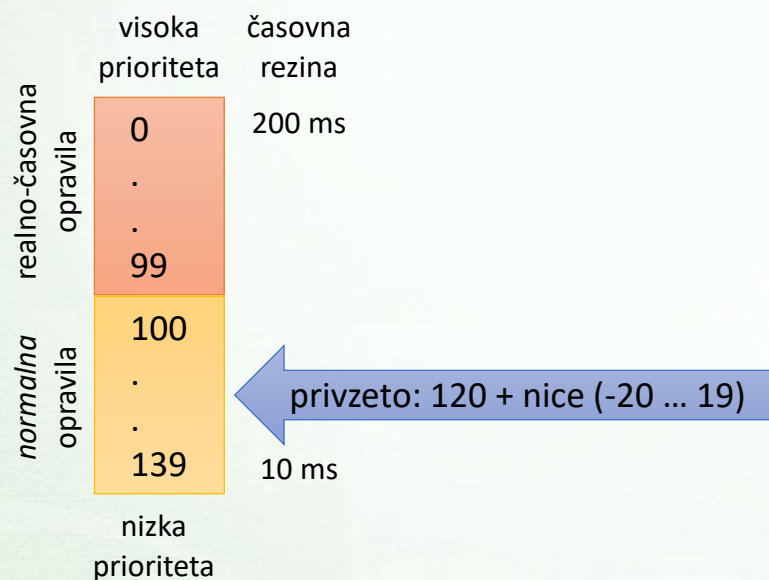
- MLFQ – večnivojska odzivna vrsta
 - pretentanje razvrščevalnika
 - če posel vrača procesor tik preden se izteče rezina, se mu prioriteta ne zmanjša
 - rešitev: natančno vodimo evidenco o preostanku časovne rezine
 - ugaševanje razvrščevalnika
 - Koliko nivojev oz. prioritet imeti?
 - Kako velika je časovna rezina za posamezno vrsto?
 - Kako pogosto povišamo prioriteto?



Praktični algoritmi

- Razvrščevalnik Linux $O(1)$

- izbira in dodajanje opravila v konstantnem času
- z odvzemanjem in prioriteto (140 prioritetnih vrst)



Odziv

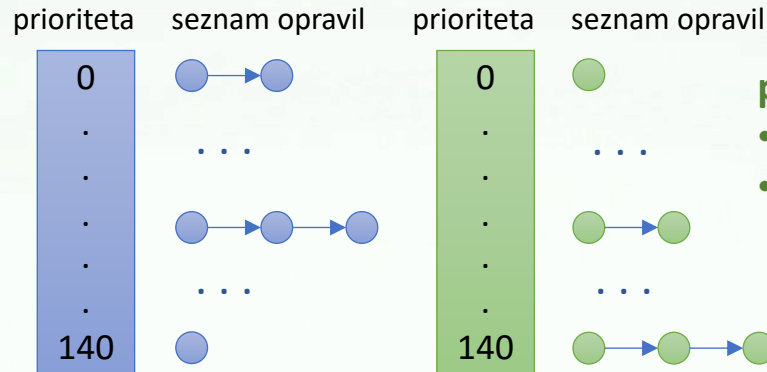
- daljše spanje
 - interaktivno opravilo
 - dvig prioritete za -5
- krajše spanje
 - CPE intenzivno
 - spust prioritete za 5

Praktični algoritmi

• Razvrščevalnik Linux $O(1)$

aktivna tabela (active array)

- uporaba za izbiro naslednjega opravila
- konstanten čas izbire in dodajanja opravila
- opravilo ostane v vrsti dokler se mu ne izteče časovna rezina



pretečena tabela (expired array)

- neaktivna pravila
- vsebuje opravila, ki so porabila časovno rezino

Ko se aktivna tabela izprazni, zamenjaj aktivno in pretečeno tabelo.

- velika časovna rezina za interaktivna opravila: dobijo več časa, vendar hitro blokirajo, zato se jim rezina ne izteče (ostajajo v aktivni tabeli)
- ko se rezine interaktivnih opravil porabijo, pridejo na vrsto CPU intenzivna opravila
- majhna časovna rezina za CPU intenzivna: se hitro izteče in opravilo gre med pretečene

Praktični algoritmi

CFS – completely fair scheduler

- CFS – popolnoma pošteno razvrščanje
 - poštenost
 - v danem časovnem intervalu naj posli dobijo delež, ki je enakovreden njihovi prioriteti
 - Linux CFS razvrščevalnik
 - izbira posla z najmanjšim porabljenim časom izvajanja (ns resolucija)
 - skrajno levo vozlišče rdeče-**črnega** drevesa 😊
 - izvedba za kvečjemu toliko časa, da ujamemo naslednji posel z najmanjšim porabljenim časom
 - po izvedbi se posel ponovno vstavi v drevo z novim skupno porabljenim časom izvajanja
 - za posle nižje prioritete čas teče hitreje