

IN1805 I – Operating System Concepten

Hoofdstuk 5: CPU scheduling

CPU scheduling : introductie

- CPU scheduling nodig bij multiprogrammering
- doel: een zo hoog mogelijke CPU-bezetting, bij tevreden gebruikers
- proces bestaat uit afwisselend CPU-bursts en I/O-bursts
- lengte van CPU-bursts exponentieel verdeeld
- CPU scheduler is short-term scheduler, selecteert een proces uit de ready queue

CPU scheduling : criteria

Mogelijke criteria:

- CPU-bezetting
- throughput
- turnaround time (tijd tussen het aanbieden van een (batch) job en het verschijnen van de output)
- waiting time (d.i. *missed time* : wel ready, geen CPU)
- respons time (tijd tussen ENTER en het verschijnen van het antwoord)
 - Stabiël is belangrijker dan gemiddeld kort maar variërend

Illustratie d.m.v. *Gantt charts*

CPU scheduling : preemption

definitie: Nonpreemptive scheduling houdt in, dat een proces dat de CPU heeft, deze mag houden totdat het de CPU vrijgeeft door te eindigen of naar de wachttoestand te gaan.

preemptive scheduling betekent dat de CPU ontnomen kan worden terwijl het proces deze nog kan gebruiken.

CPU-scheduling algoritmen:

FCFS

First Come First Served (FCFS),
ook genoemd First In, First Out (FIFO)

- nonpreemptive
- voordeel: eenvoudige implementatie
- nadelen:
 - voor kleine hoeveelheid tijd relatief lang wachten
 - konvooi effect

CPU-scheduling algoritmen

SJF

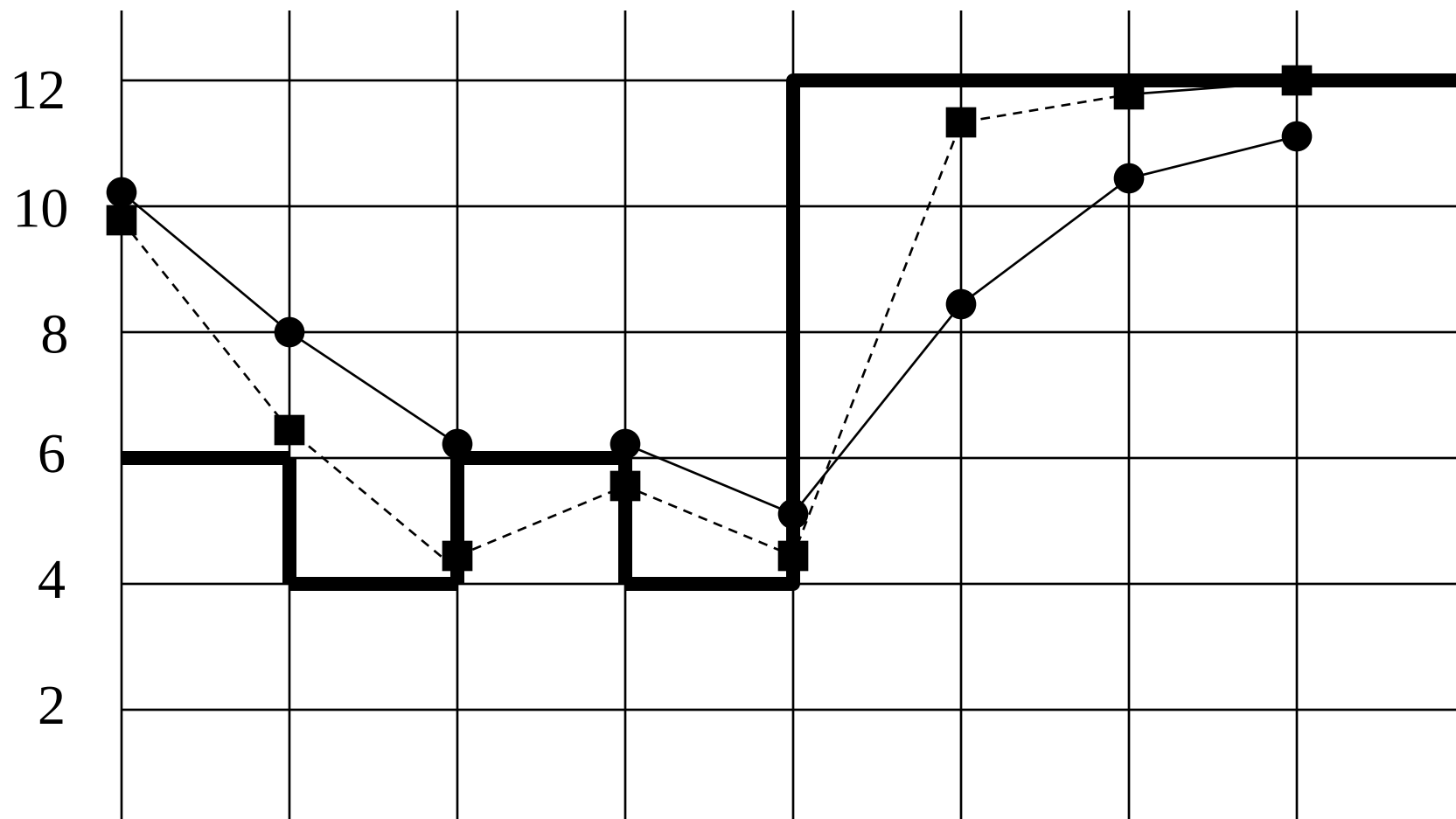
Shortest Job First (SJF),

eigenlijk Shortest- *CPU-burst* -First

- preemptive of nonpreemptive
- preemptive SJF geeft bewijsbare de kortst mogelijke missed time
- probleem: voorspellen van lengte op basis van exponentieel gemiddelde:

$$T_{n+1} = a t_n + (1 - a) T_n$$

T_i = voorspelling, t_j = gerealiseerde tijd



t	6	4	6	4	12	12	12	12	
a=0.5 T	10	8	6	6	5	8.5	10.25	11.12	●
a=0.9 T	10	6.40	4.24	5.82	4.18	11.22	11.92	11.99	■

3-7

CPU-Scheduling algoritmen

Priority Scheduling

- Wijst CPU toe op grond van prioriteit
- prioriteit kan zijn :
 - interne prioriteit (b.v op basis van gebruik van resources)
 - externe prioriteit (b.v. contract, of gebruiker is VIP)
- preemptive of nonpreemptive
- probleem:
 - kans op starvation, d.w.z. bepaalde processen komen nooit aan de beurt.
 - “oplossing”: aging

CPU-scheduling algoritmen

Round Robin (1)

- Speciaal voor Timesharing systemen
- preemptive
- gebruikers krijgen om de beurt een tijdquantum q (timeslice)
- als het proces binnen de timeslice niet verder kan, gaat de CPU naar het volgende proces.
- Als de CPU-burst aan het einde van de timeslice nog niet is afgelopen, komt er een interrupt en gaat de CPU ook naar het volgende proces.

CPU-scheduling algoritmen

Round Robin (2)

Keuze van q is erg belangrijk

- als q erg groot is, dan lijkt RR op FCFS
- als q klein is : processor sharing
- Context switchen kost tijd, als q te klein is : thrashing
- richtlijn: q moet zo groot zijn, dat de meerderheid van de CPU-bursts in 1 time slice kan worden afgehandeld.

CPU-scheduling algoritmen

ML

Multilevel queue scheduling: hanteert verschillende queues voor verschillende soorten werk, b.v.

- foreground (interactief)
- background (batch)
- voor verschillende queues, verschillende algoritmen
- tijd tussen de queues te verdelen op basis van:
 - prioriteit
 - timeslices

CPU-scheduling algoritmen

MLFB

- Multi-Level scheduling met Feed Back, maakt het mogelijk dat prcoessen van queue kunnen wisselen.
- parameters:
 - aantal queues
 - verdeling tijd tussen de queues
 - methode voor demoveren (b.v. overschrijding tijdlimiet)
 - methode voor promoveren (b.v. aging)

MLFB is het meest universele scheduling algoritme,
probleem : keuze van de parameters.

Multiprocessor scheduling

- aannamen:
 - homogene processoren
 - uniform memory access
- scheduling d.m.v. één READY-queue, mogelijkheden:
 - iedere processor zelf schedulend (*symmetrische multiprocessing*)
 - scheduling door één processor (*master/slave* structuur, *asymmetrische multiprocessing*)

CPU-scheduling algoritmen:

Algoritme-evaluatie overzicht

- Eerst criteria vaststellen, dan algoritme evalueren volgens een bepaalde methodiek.
 - Analytische evaluatie
 - deterministisch evalueren
 - queueing modellen
 - Simulatie
 - random input overeenkomstig mathematische verdeling
 - random input overeenkomstig gemeten verdeling
 - input van een trace tape
 - Testen in de praktijk

Deterministische evaluatie

- Bekijkt het gedrag van algoritmen bij exact bekende workload
- Geeft nauwkeurige resultaten, maar deze zijn alleen geldig voor deze workload
- is nuttig om inzicht te krijgen (trends te kunnen ontdekken)
- is te specifiek om hierop de keus van parameters te kunnen baseren.

Queueing model

- Wachtrij theorie gaat uit van
 - model van servers en wachtrijen
 - verdeling CPU-bursts (algemeen: service times)
 - verdeling tussen-aankomsttijden (inter-arrival times)
 - scheduling policies
- berekent: lengte van de wachtrij, wachttijden, etc.
- Formule van Little: $n = \lambda \times W$
 - n = gemiddelde rijlengte
 - λ = gemiddelde aantal aankomsten per seconde
 - W = gemiddelde wachttijd in de rij

Geldig voor iedere verdeling van aankomsttijden en service tijden en ieder scheduling algoritme

Simulatie

- onderdelen:
 - Een model
 - gedrag van de componenten geprogrammeerd in software
 - software bevat ook een klok
 - invoer
- werkwijze:
 - bij iedere wijziging van de klok is een toestandsovergang mogelijk
 - toestandsovergangen mede bepaald door invoer
 - tijden, input, toestandsovergangen en toestanden worden geregistreerd en gepresenteerd
- nadelen:
 - gedetailleerd model maken is moeilijk
 - gedetailleerde simulatie is zeer rekenintensief

voorbeeld Linux

(tot en met versie 2.4)

Twee scheduling algoritmen

- Time sharing: eerlijke preemptive scheduler,
 - gebaseerd op prioriteit en krediet
 - bij keuze gaat de CPU naar het proces met het hoogste krediet
 - bij iedere timer interrupt wordt krediet van running proces 1 minder
 - bij krediet 0 CPU naar ander proces
 - als geen runnable proces nog krediet heeft, volgt herkrediterings operatie: voor **ieder** process: $\text{krediet} = \text{krediet}/2 + \text{prioriteit}$
- Real time (soft)
 - FCFS + prioriteit (nonpreemptive), of
 - RR + prioriteit (preemptive)