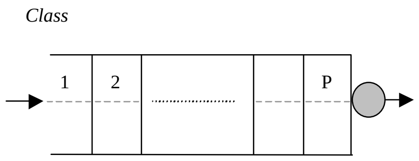
Priority Operation

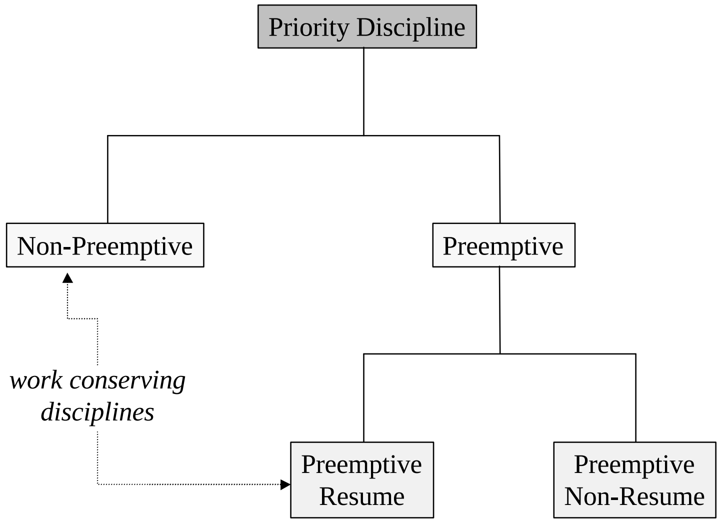
of

The M/G/1 Queue



Class 1 Lowest Priority Class

Head of Line (HOL) Priority Operation of M/G/1 Queue



Priority Discipline

Non-Preemptive

Preemptive

work conserving disciplines

Preemptive Resume

Preemptive Non-Resume

Non-Preemptive Priority

•Consider an arrival of priority class *j* when the server is serving a job of lower priority class *k, j>k*.

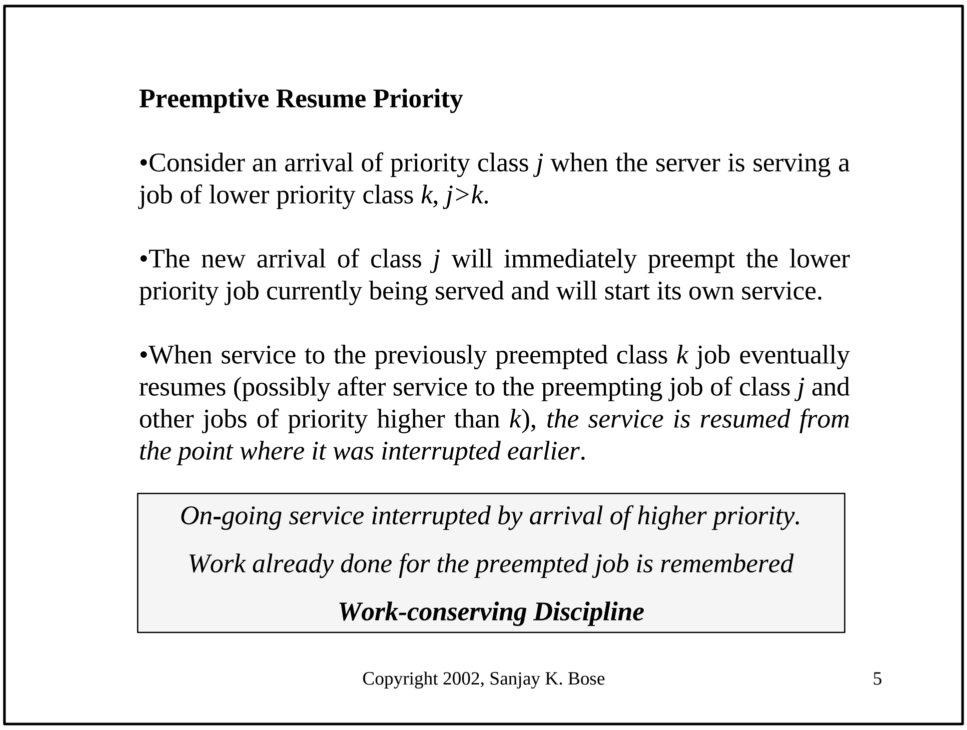
•The new arrival, in spite of being of a priority level higher than the current job in service, will not interrupt the on-going service.

•Instead, it will join the queue (FCFS) at the end of the queue of its own priority class, i.e. Class *j*, and wait for the current job to finish service.

•Normal HOL priority operation will resume once the on-going service is over

On-going service is not interrupted, even if there are new arrivals of higher priority

Work-conserving Discipline



Preemptive Resume Priority

- Consider an arrival of priority class *j* when the server is serving a job of lower priority class *k*, *j*>*k*.

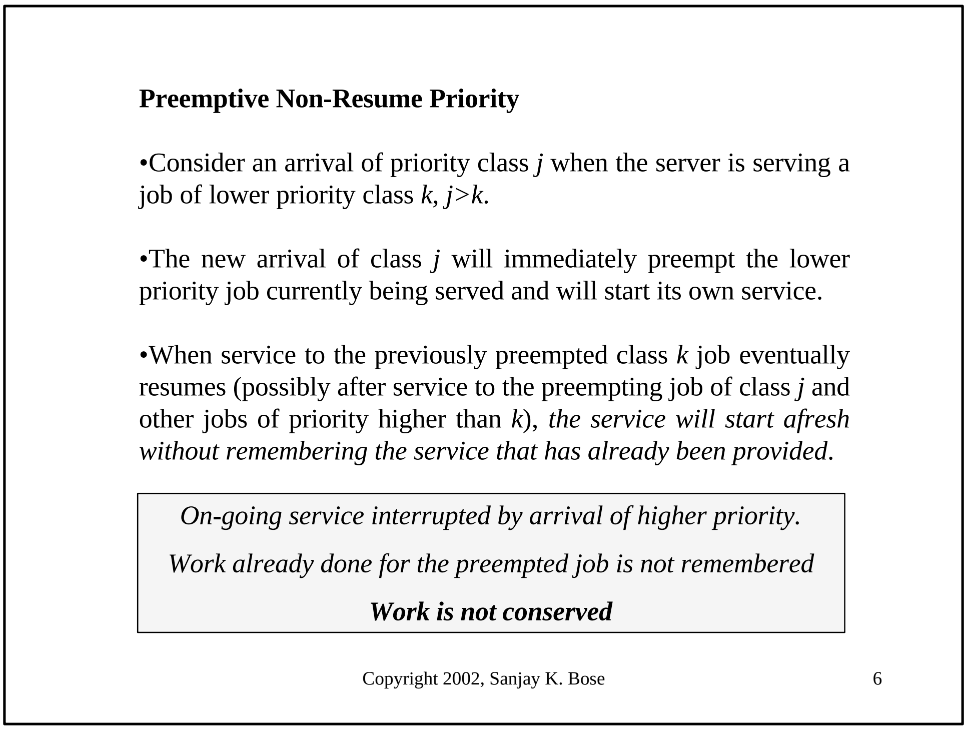
- The new arrival of class *j* will immediately preempt the lower priority job currently being served and will start its own service.

- When service to the previously preempted class *k* job eventually resumes (possibly after service to the preempting job of class *j* and other jobs of priority higher than *k*), *the service is resumed from the point where it was interrupted earlier*.

On-going service interrupted by arrival of higher priority.

Work already done for the preempted job is remembered

Work-conserving Discipline



Preemptive Non-Resume Priority

- Consider an arrival of priority class *j* when the server is serving a job of lower priority class *k*, *j*>*k*.

- The new arrival of class *j* will immediately preempt the lower priority job currently being served and will start its own service.

- When service to the previously preempted class *k* job eventually resumes (possibly after service to the preempting job of class *j* and other jobs of priority higher than *k*), *the service will start afresh without remembering the service that has already been provided*.

On-going service interrupted by arrival of higher priority.

Work already done for the preempted job is not remembered

Work is not conserved

- Arrival Process for Class *i* is Poisson with rate l_i *i=1,….,P*

- Arrival Processes of different classes independent of each other

- The overall arrival process will also be Poisson with rate l

$$l = \sum_{i=1}^{P} l_i$$

Service time for Class *i* has mean $\overline{X}_i$ and second moment $\overline{X_i^2}$ with pdf $b_i(t)$, cdf $B_i(t)$ and L.T. of the pdf as $L_{bi}(s)$

Service times for the different classes assumed to be independent of each other

Traffic of priority class *i* $r_i = l_i \overline{X}_i$ *i=1,…..,P*

Total Traffic $r = \sum_{i=1}^{P} r_i = l\overline{X}$

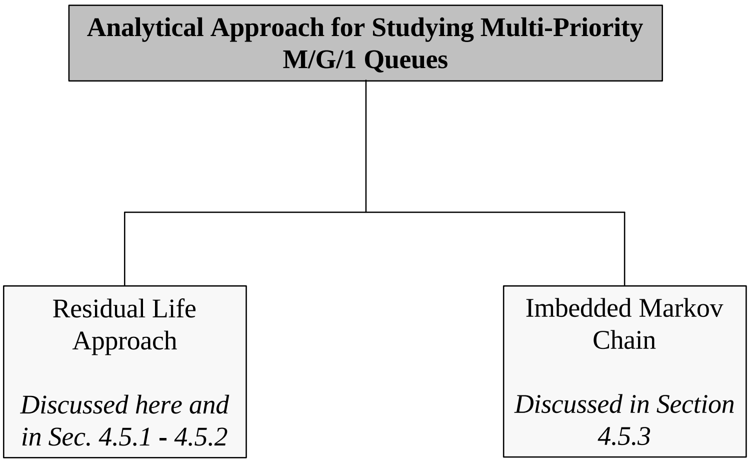
where $\overline{X} = \sum_{i=1}^{P} \frac{l_i}{l} \overline{X}_i$ is the *mean overall service time*

Condition for the P-Priority M/G/1 Queue to be Stable

$$r = \sum_{i=1}^{P} r_i < 1$$

For Work-Conserving Queueing Disciplines

For multi-priority queues, it is possible for the queue to become unstable for lower priority traffic while still being stable for the higher priorities.



Analytical Approach for Studying Multi-Priority M/G/1 Queues

Residual Life Approach

Discussed here and in Sec. 4.5.1 - 4.5.2

Imbedded Markov Chain

Discussed in Section 4.5.3

Also see the additional notes for analytical approaches for Finite Capacity, Multi-Server, Multi-Priority M/M/n/K type Queues

Residual Life Analysis for a Non-Preemptive Priority M/G/1 Queue

Number of Priority Classes = P (Class 1 lowest priority)

N_{qk} Number of class k jobs waiting in queue (prior to service)

W_{qk} Mean waiting time in queue for jobs of priority class k

$N_{qk} = l_k W_{qk}$ (Little's Result for class k jobs)

R Mean Residual Service Time for job currently being served when an arrival (of any priority class) occurs

$$R = \frac{1}{2} \sum_{i=1}^{P} l_i \overline{X_i^2} \qquad (4.40)$$

We now consider each priority class separately, starting with the highest priority class P and ending with the lowest priority class 1

Class P

$$W_{qP} = R + \overline{X}_P N_{qP}$$

leading to

$$W_{qP} = \frac{R}{1 - r_P} \qquad (4.41)$$

Class P-1

$$W_{q(P-1)} = R + \overline{X}_P N_{qP} + \overline{X}_{P-1} N_{q(P-1)} + \overline{X}_P l_P W_{q(P-1)}$$

leading to

$$W_{q(P-1)} = \frac{R}{(1-r_P)(1-r_P-r_{P-1})} \qquad (4.44)$$

Copyright 2002, Sanjay K. Bose

Class P-2

$$\begin{aligned}W_{q(P-1)} &= R + \overline{X}_P N_{qP} + \overline{X}_{P-1} N_{q(P-1)} + \overline{X}_{P-2} N_{q(P-2)} \\ &+ \overline{X}_P l_P W_{q(P-2)} + \overline{X}_{P-1} l_{P-1} W_{q(P-2)}\end{aligned}$$

leading to

$$W_{q(P-2)} = \frac{R}{(1-r_P-r_{P-1})(1-r_P-r_{P-1}-r_{P-2})} \qquad (4.46)$$

Copyright 2002, Sanjay K. Bose

Therefore, in general, we will get

$$W_{qP} = \frac{R}{1 - r_P} \qquad i=P$$

$$W_{q(P-i)} = \frac{R}{(1-\sum_{j=0}^{i-1} r_{P-j})(1-\sum_{j=0}^{i} r_{P-j})} \qquad i=1,\ldots..,(P-1)$$

(4.47)

$$W_i = W_{qi} + \overline{X}_i \qquad i=1,\ldots..,(P-1) \qquad (4.48)$$

The parameters N_i and N_{qi} may then be found using Little’s Result

Residual Life Analysis for a Preemptive Resume Priority M/G/1 Queue

- Consider *P* priority classes as before with class *P* of highest priority
- Jobs of priority classes *1,….., (P-1)* may be interrupted by the arrival of new jobs with higher priority
- No loss of work as interrupted job resumes service from point of interruption
- Queueing Delay can be meaningfully defined only for Class *P*. For the lower priority classes, this parameter will not be important as a job’s service can be interrupted even after it starts service
- The Residual Service Time seen by an arrival will depend on the class of the new arrival

R_k = Mean Residual Service Time as seen by a new job arrival of class k

$$R_k = \sum_{i=k}^{P} \frac{1}{2} \lambda_i \overline{X_i^2} \qquad k=1,\ldots,P \qquad (4.49)$$

- Note that, as mentioned earlier, R_k depends on the class of the new arrival.
- An arrival of the highest priority class will see the smallest mean residual service time as it will preempt any ongoing service of priority class other than itself.
- Arrivals of lower priority class will only be able to preempt jobs of priority lower than themselves

Class P

In this case, we can define a mean queueing delay W_{qP} as before

$$W_{qP} = R_P + \overline{X}_P N_{qP} \quad \Longrightarrow \quad W_{qP} = \frac{R_P}{1-\rho_P} \qquad (4.50)$$

$$\Downarrow$$

$$W_P = W_{qP} + \overline{X}_P = \frac{\overline{X}_P(1-\rho_P)+R_P}{(1-\rho_P)} \qquad (4.51)$$

Mean Total Delay for Class P

Class P-1

$$W_{P-1} = \overline{X}_{P-1} + \frac{R_{P-1}}{1 - r_P - r_{P-1}} + \overline{X}_P \, l_P W_{P-1} \qquad (4.52)$$

See Section 4.5.2 for the arguments justifying this term

This leads to

$$W_{P-1} = \frac{\overline{X}_{P-1}(1 - r_P - r_{P-1}) + R_{P-1}}{(1 - r_P)(1 - r_P - r_{P-1})} \qquad (4.53)$$

Mean Total Delay for Class P-1

In general, we will get

$$W_P = \frac{\overline{X}_P(1 - r_P) + R_P}{(1 - r_P)}$$

for Class *P*

$$W_k = \frac{\overline{X}_k(1 - r_P - \ldots\ldots - r_k) + R_k}{(1 - \ldots\ldots - r_{k-1})(1 - r_P - \ldots\ldots - r_k)}$$

for Class *k*, *1£ k£ P-1*

as the total mean delay for each class of customers

Analysis of Multi-Priority M/G/1 Queue using the Imbedded Markov Chain Approach

- Though it is possible to do an analysis using this approach for the work-conserving priority disciplines, this is much more difficult than the way the mean performance results were obtained using a Residual Life Approach

- See Section 4.5.3 for the analysis of a *2-Priority M/G/1 Queue* following this approach.