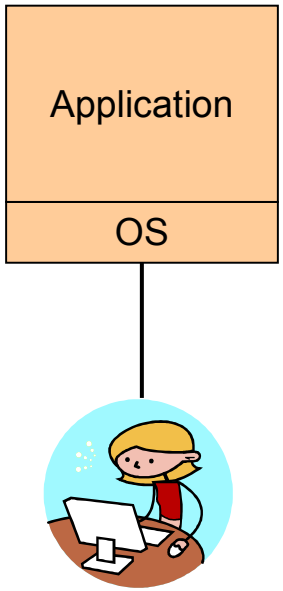
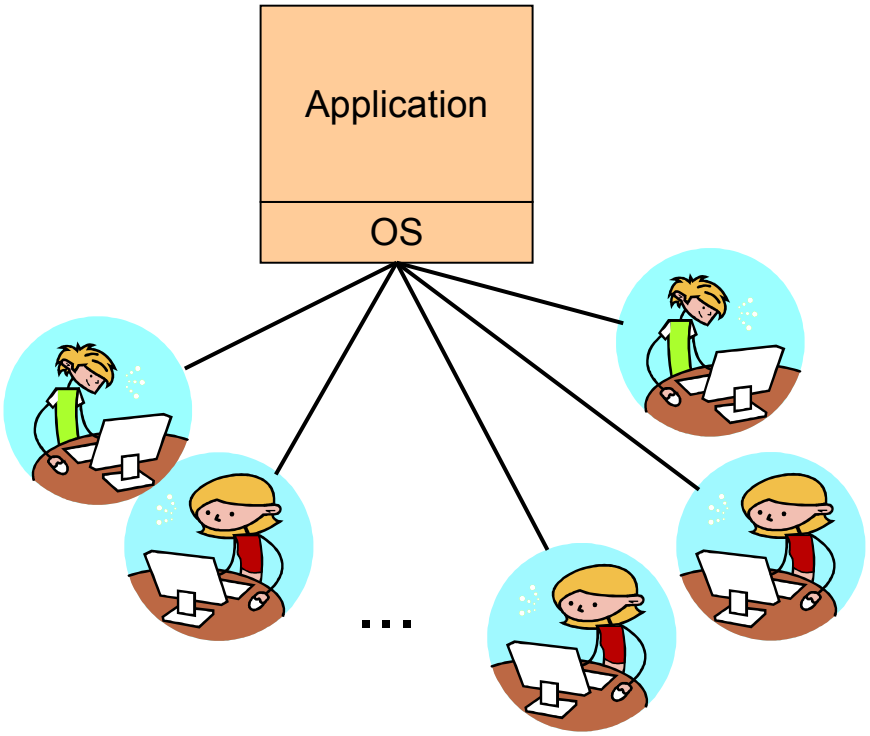
What is Work Management?

More User

Single User Enviroment

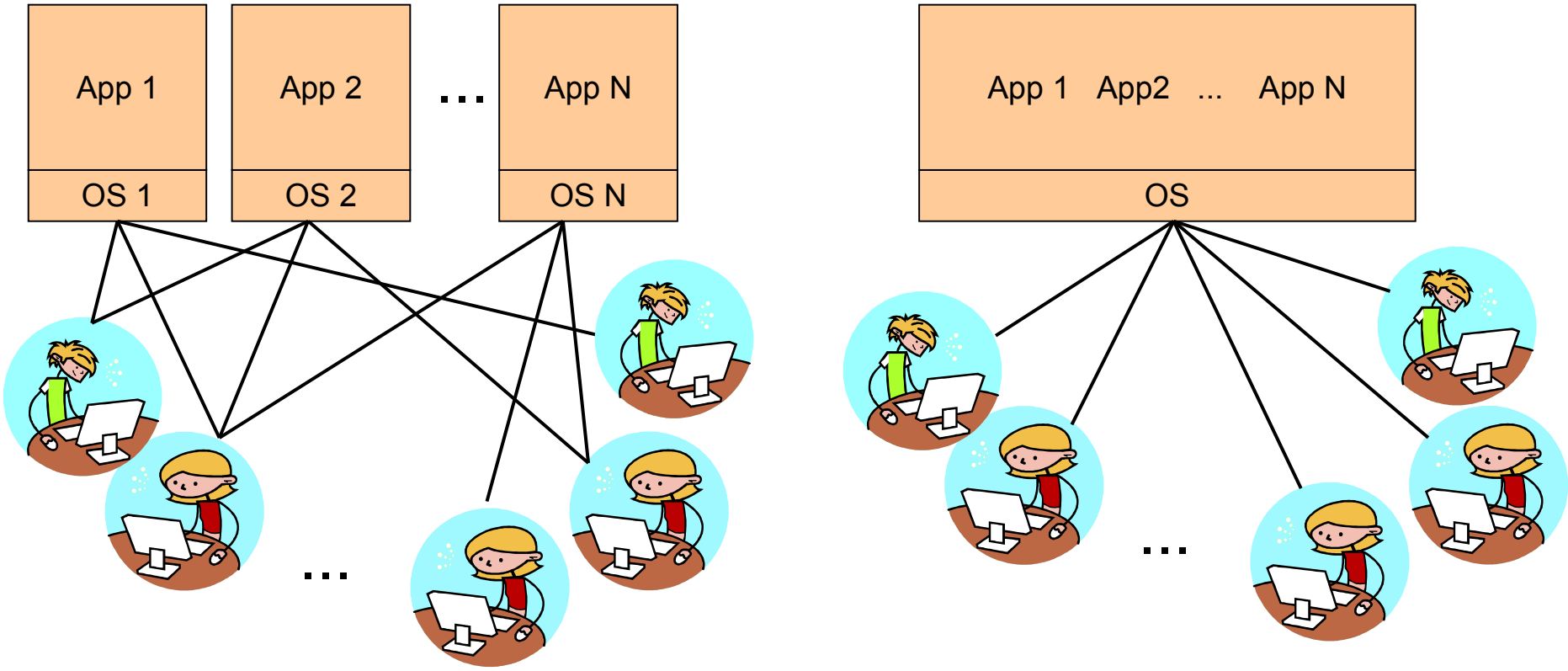


Multi User Enviroment



Performance? Security? Availability?

More Applications



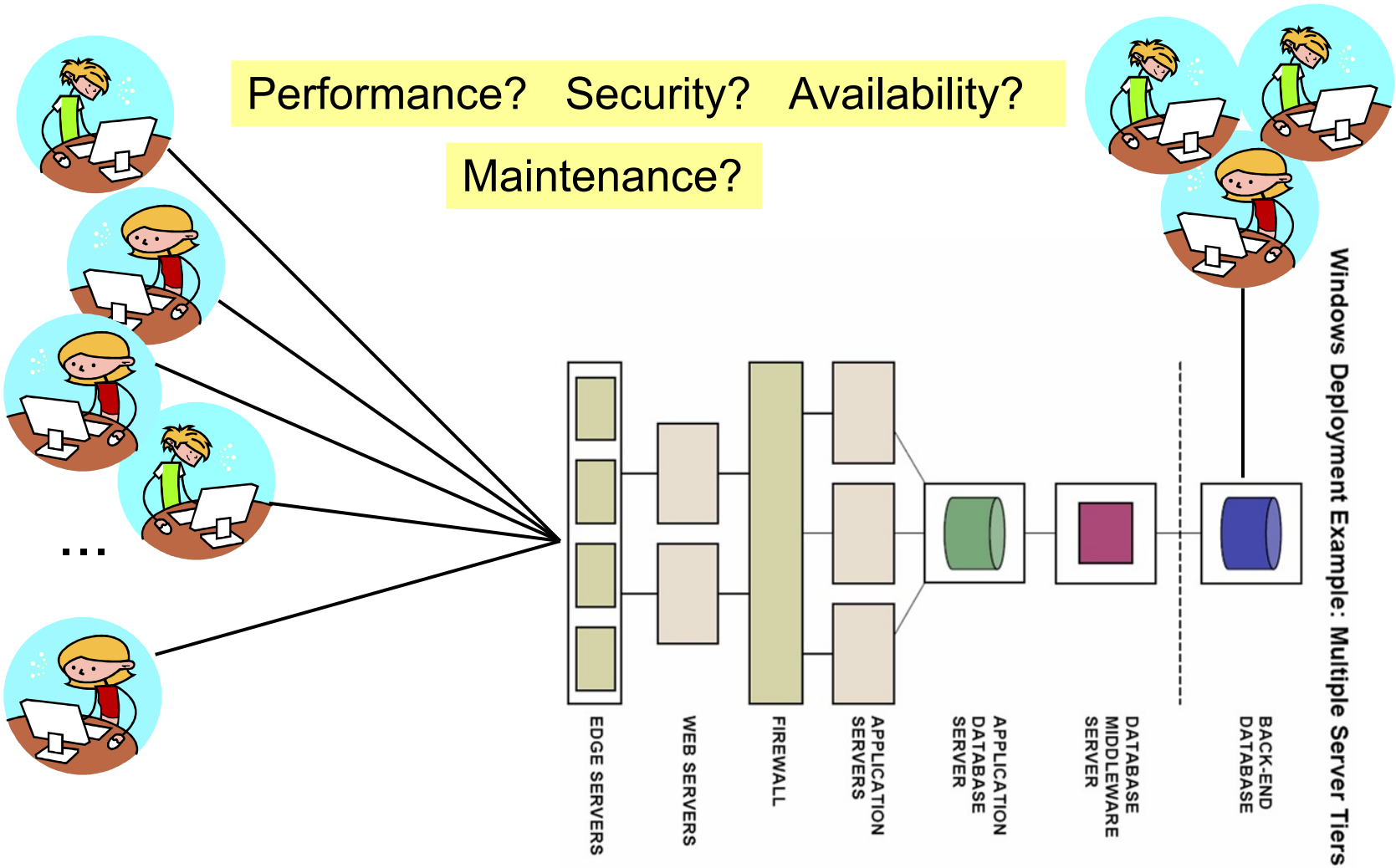
Performance? Security? Availability?

Maintenance?

Reality is more complex!

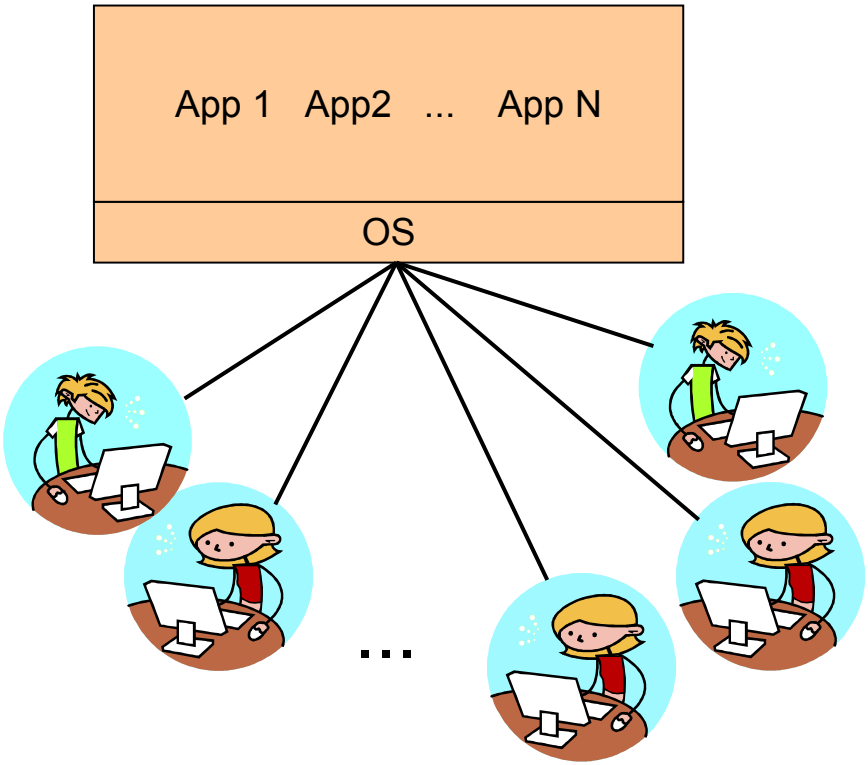
Performance? Security? Availability?

Maintenance?



Work Management

Running Multiple Applications in a single OS Instance



Performance
(or how to share ressources)

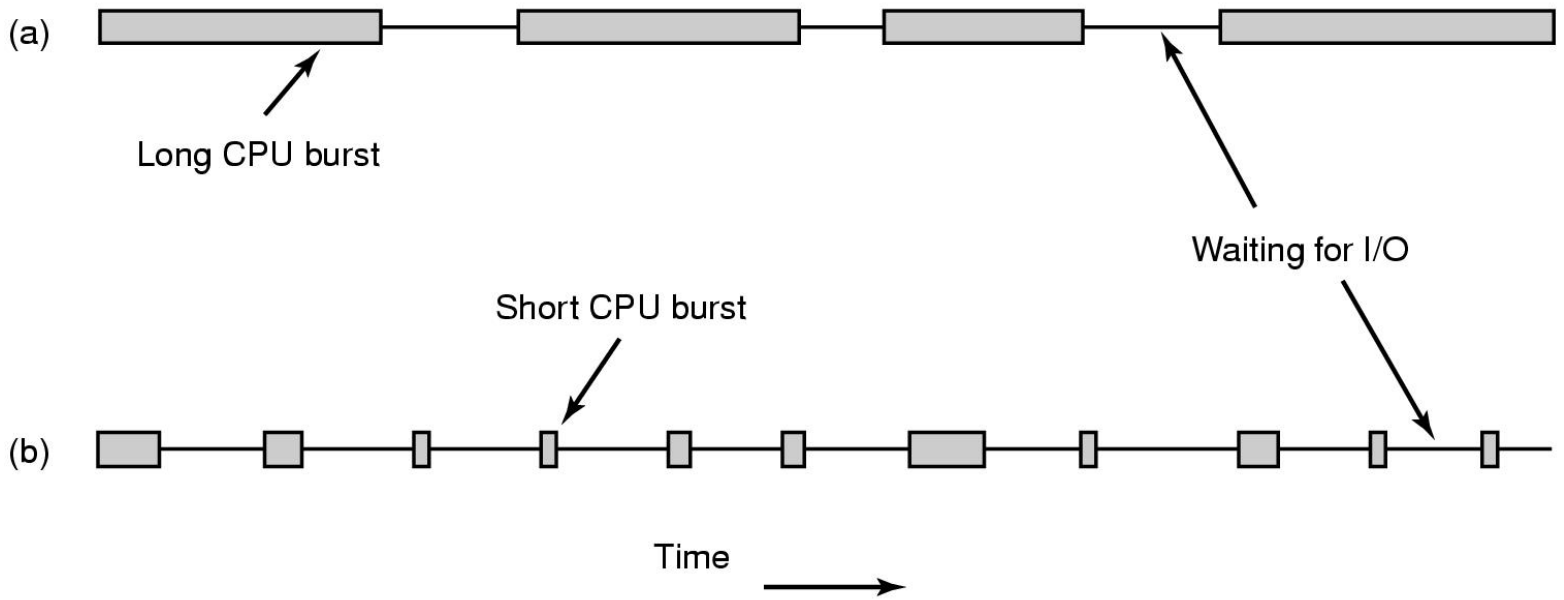
Security → Lecture 4

Availability

Do not confuse work management and LPAR!

Process Scheduling

Introduction to Scheduling



- Bursts of CPU usage alternate with periods of I/O wait
  - a CPU-bound process
  - an I/O bound process

Scheduling? Time slice?
Memory management?

Source: Andrew S. Tanenbaum: Modern Operating Systems 2th Edition

Scheduling Algorithm Goals

All systems

- Fairness - giving each process a fair share of the CPU
- Policy enforcement - seeing that stated policy is carried out
- Balance - keeping all parts of the system busy

Batch systems

- Throughput - maximize jobs per hour
- Turnaround time - minimize time between submission and termination
- CPU utilization - keep the CPU busy all the time

Interactive systems

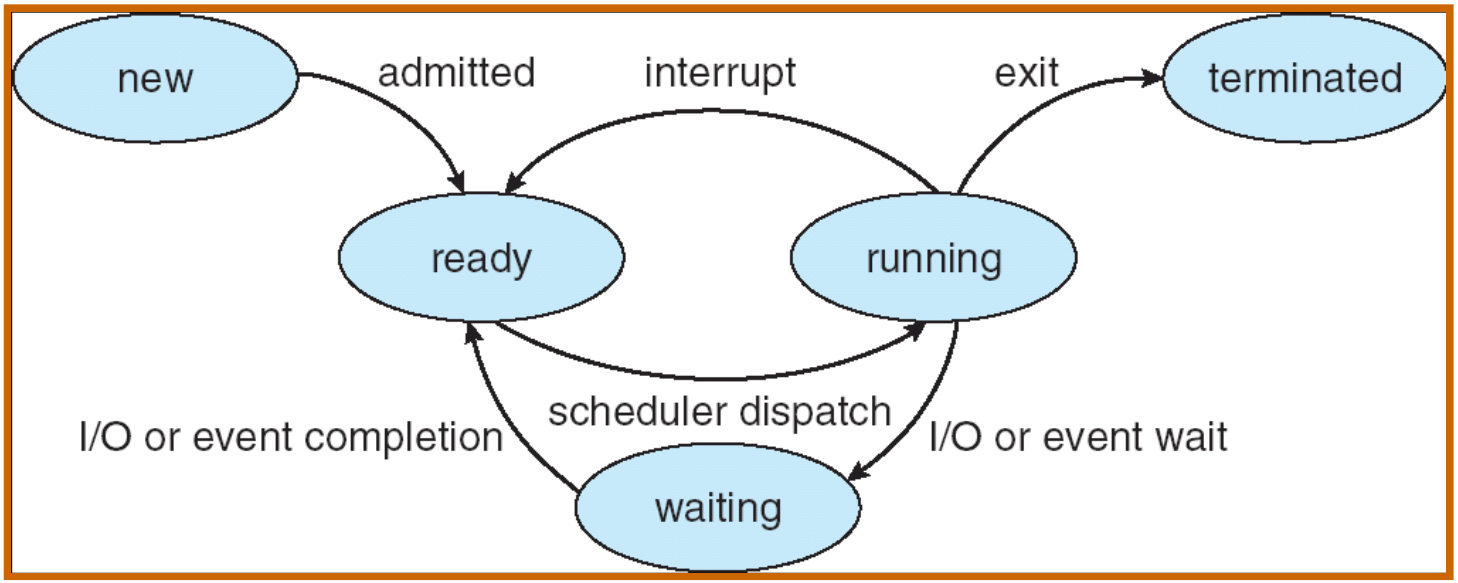
- Response time - respond to requests quickly
- Proportionality - meet users' expectations

Real-time systems

- Meeting deadlines - avoid losing data
- Predictability - avoid quality degradation in multimedia systems

Source: Andrew S. Tanenbaum: Modern Operating Systems 2th Edition

Diagram of Process State



- As a process executes, it changes *state*
  - new: The process is being created
  - running: Instructions are being executed
  - waiting: The process is waiting for some event to occur
  - ready: The process is waiting to be assigned to a processor
  - terminated: The process has finished execution

Source: Silberschatz, Galvin and Gagne: Operating System Concepts - 7th Edition

Preemptive and Non-preemptive Scheduling

- CPU scheduling decicions may take place, if a process
  - (1) switches from running to waiting state
  - (2) switches from running to ready state
  - (3) switches from waiting to ready state
  - (4) terminates
- When scheduling take place only under (1) or (4) the scheduling scheme is non-preemptive or preemptive.
- Examples:
  - non-preemptive: Windows 3.1
  - preemptive: Windows 95 or later, Mac OS X

Source: Andrew S. Tanenbaum: Modern Operating Systems 2th Edition

Context Switch

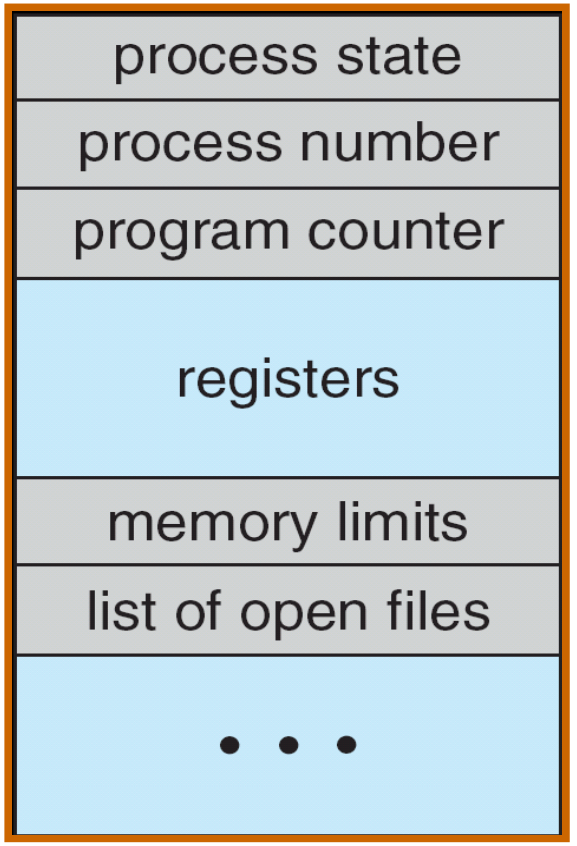
- When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process
- Context-switch time is overhead; the system does no useful work while switching
- Time dependent on hardware support

Source: Silberschatz, Galvin and Gagne: Operating System Concepts - 7th Edition

Process Control Block (PCB)

•Information associated with each process:

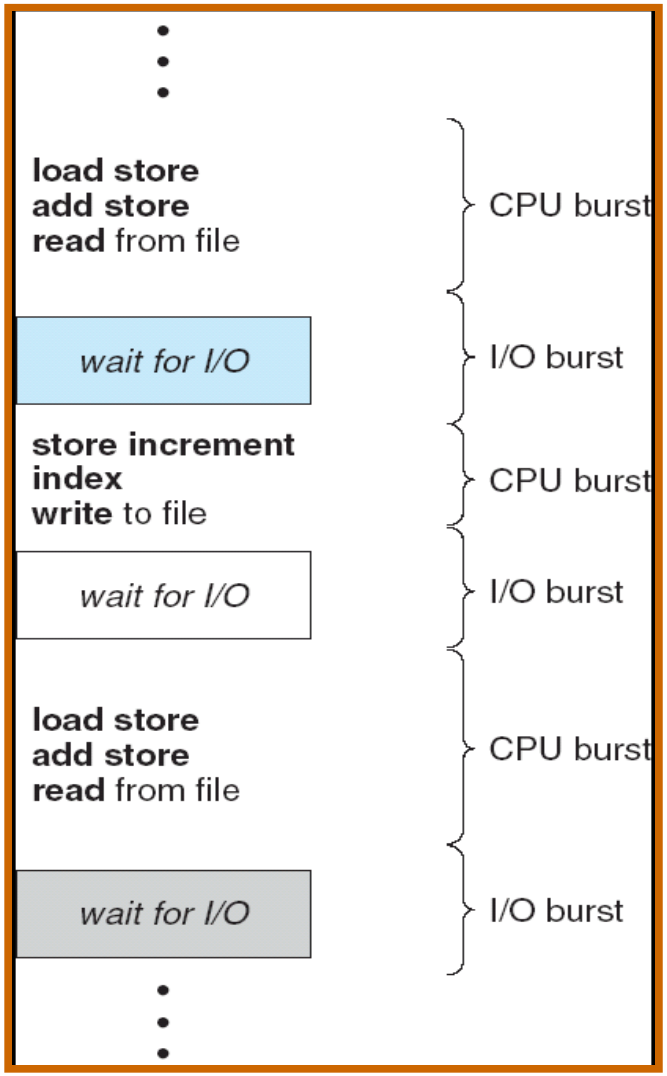
- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information



Source: Silberschatz, Galvin and Gagne: Operating System Concepts - 7th Edition

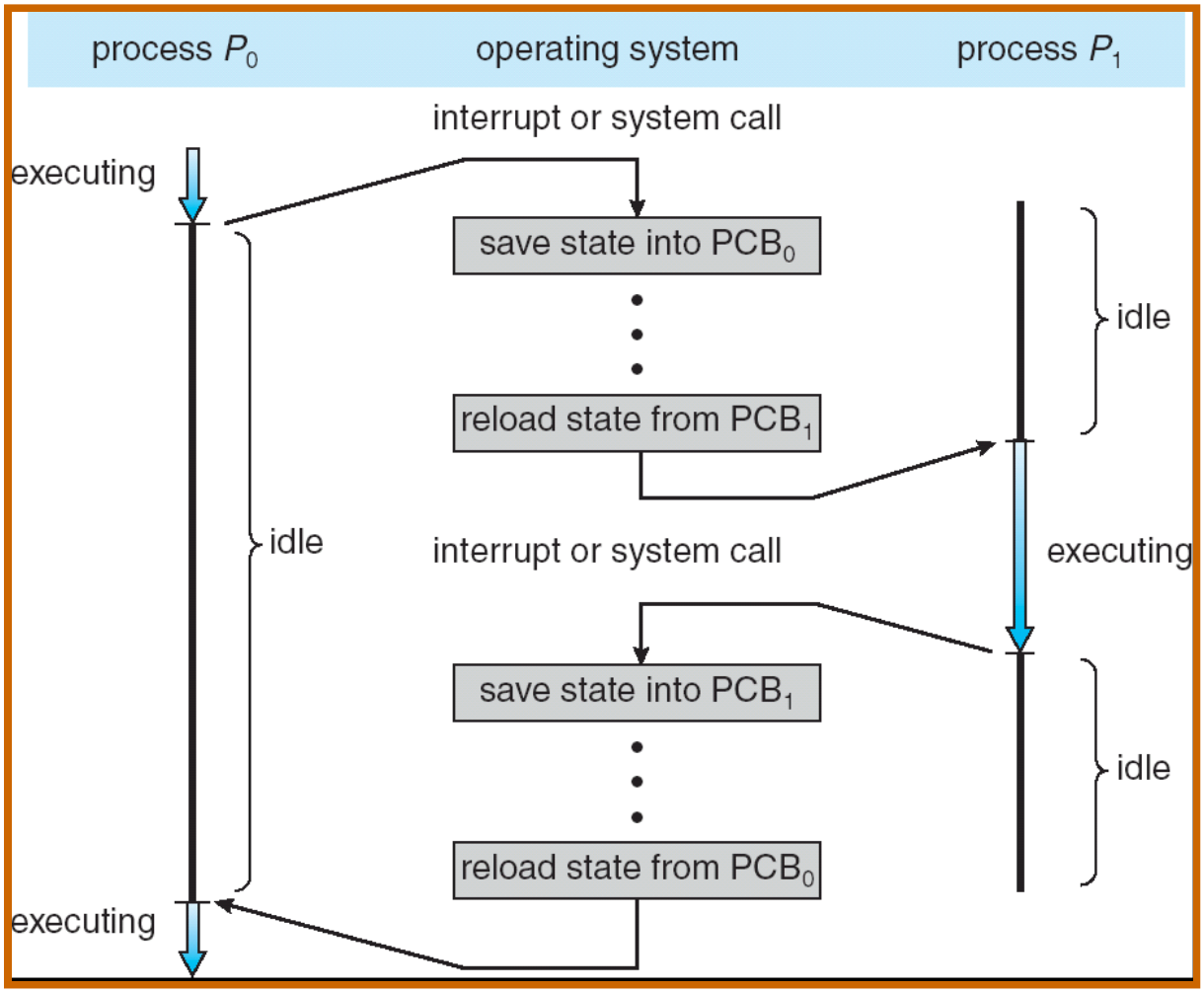
Single Process Execution

- CPU–I/O Burst Cycle – Process execution consists of a *cycle* of CPU execution and I/O wait
- CPU burst distribution



Source: Silberschatz, Galvin and Gagne: Operating System Concepts - 7th Edition

CPU Switch From Process to Process



Source: Silberschatz, Galvin and Gagne: Operating System Concepts - 7th Edition

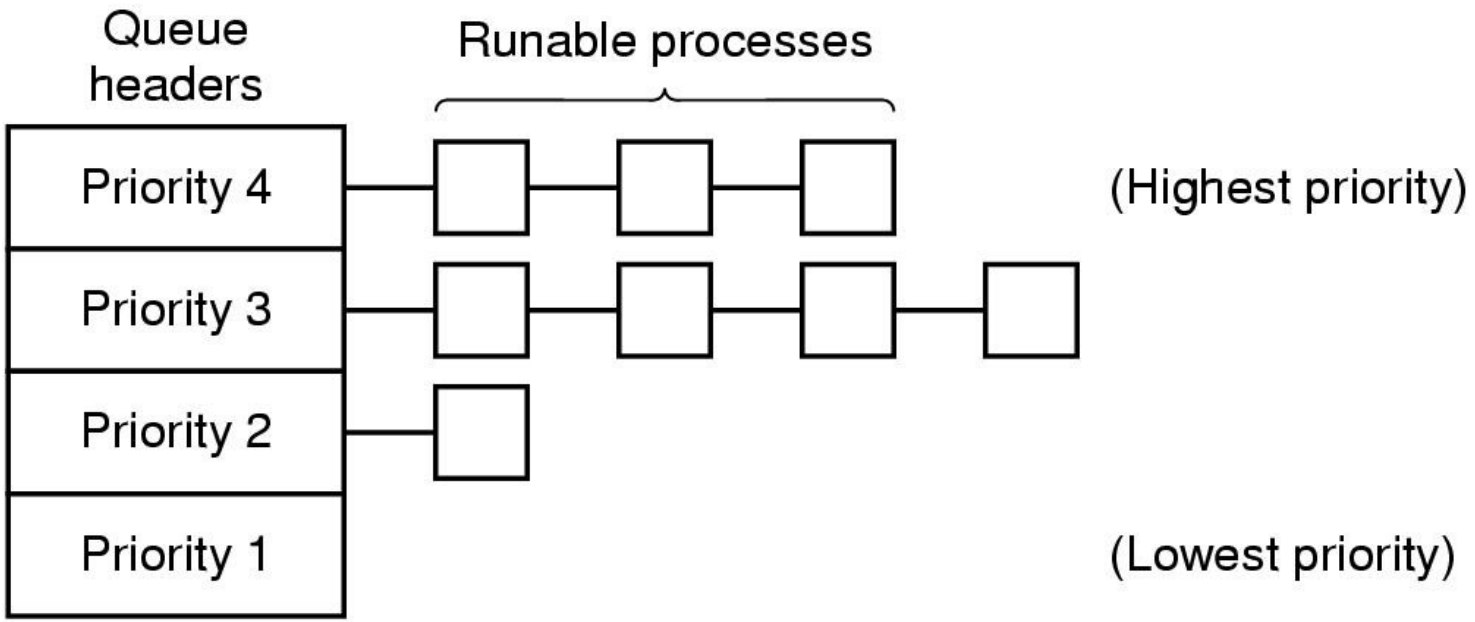
Efficient process scheduling?

Scheduling Criteria

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – # of processes that complete their execution per time unit
- **Turnaround time** – amount of time to execute a particular process
- **Waiting time** – amount of time a process has been waiting in the ready queue
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, **not** output (for time-sharing environment)

Source: Silberschatz, Galvin and Gagne: Operating System Concepts - 7th Edition

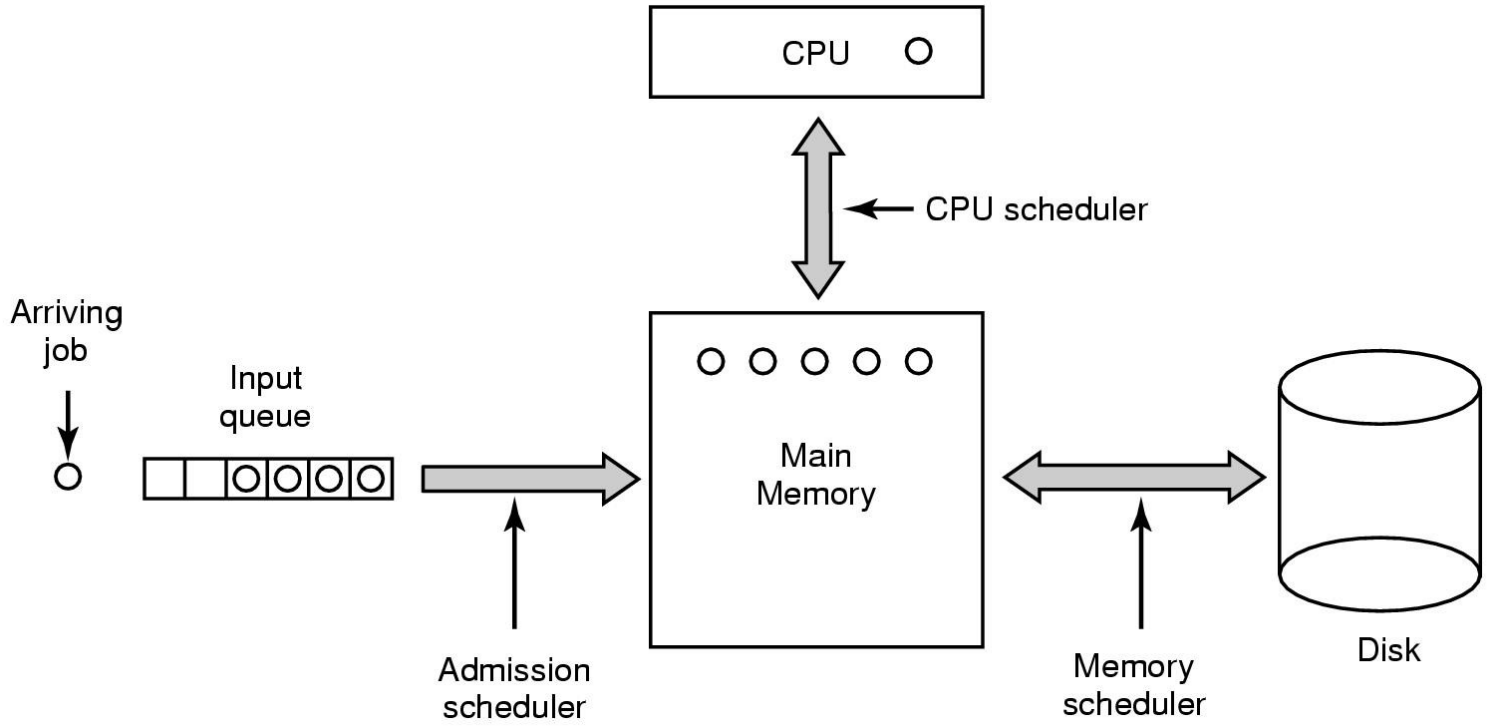
Scheduling in Interactive Systems



A scheduling algorithm with four priority classes

Source: Andrew S. Tanenbaum: Modern Operating Systems 2th Edition

Scheduling in Batch Systems



Three level scheduling

Source: Andrew S. Tanenbaum: Modern Operating Systems 2th Edition

Example: IBM i

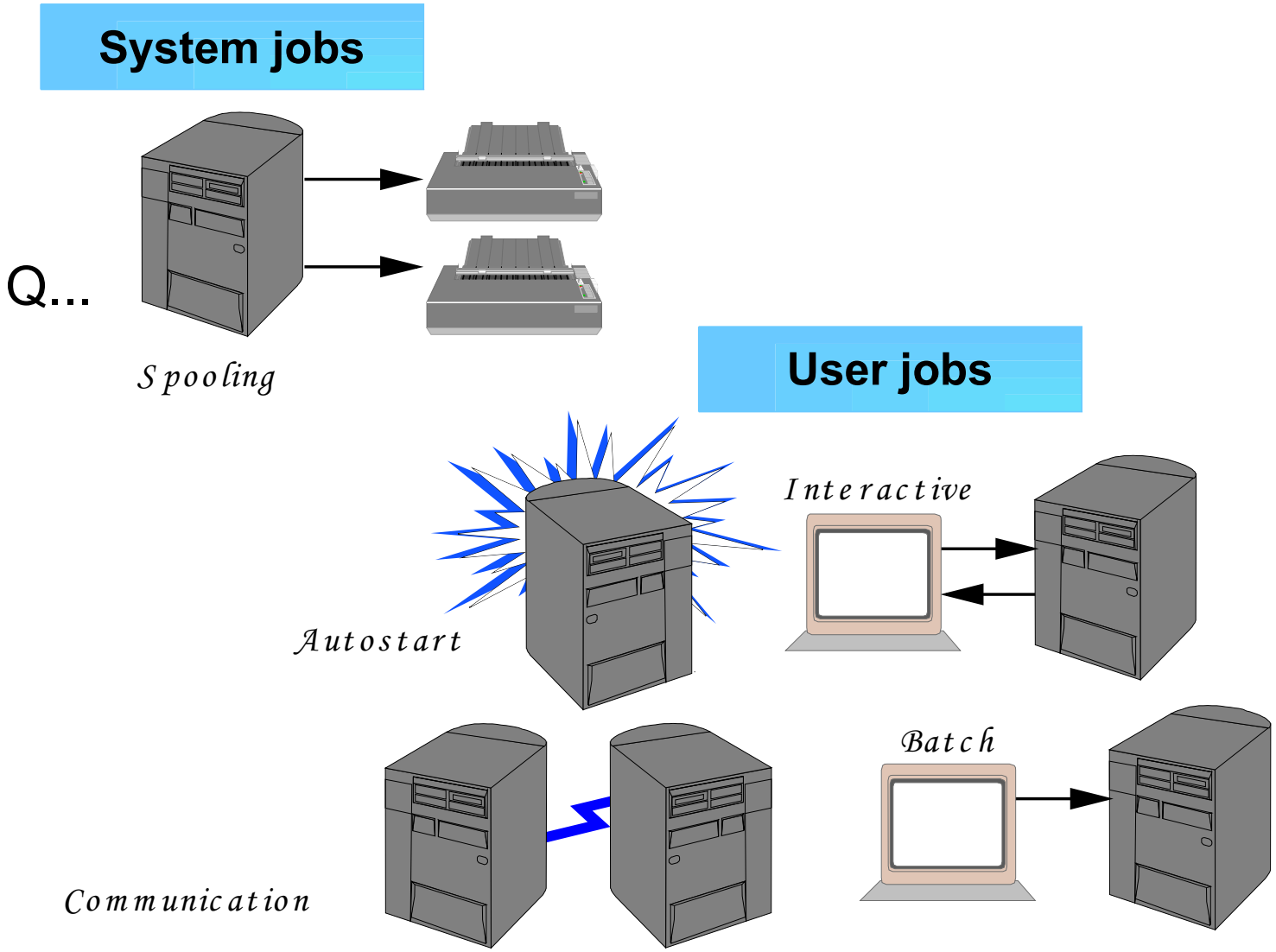
aka i5/OS, OS/400

5.1 Operator‘s View

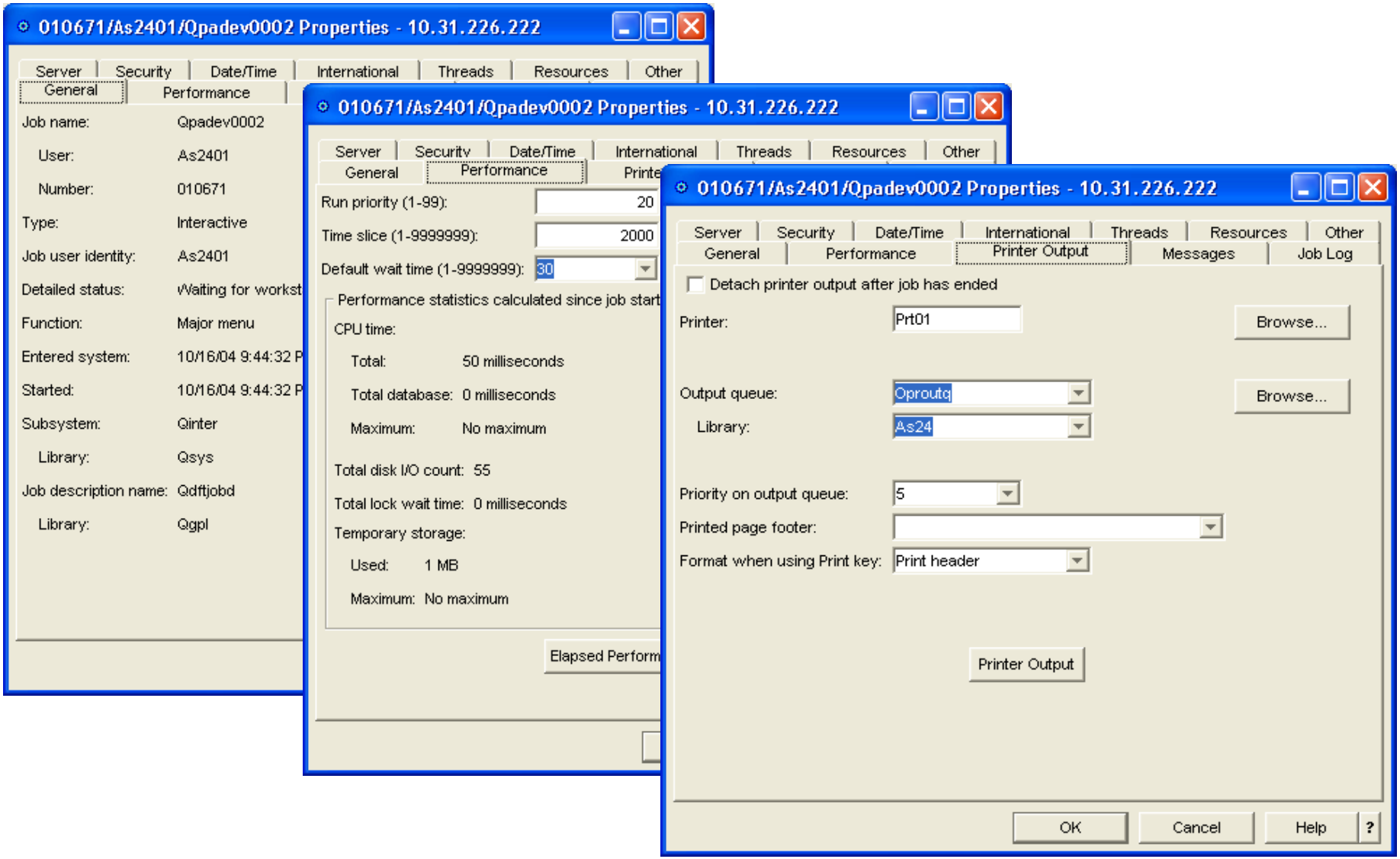
Job

Qualified job name

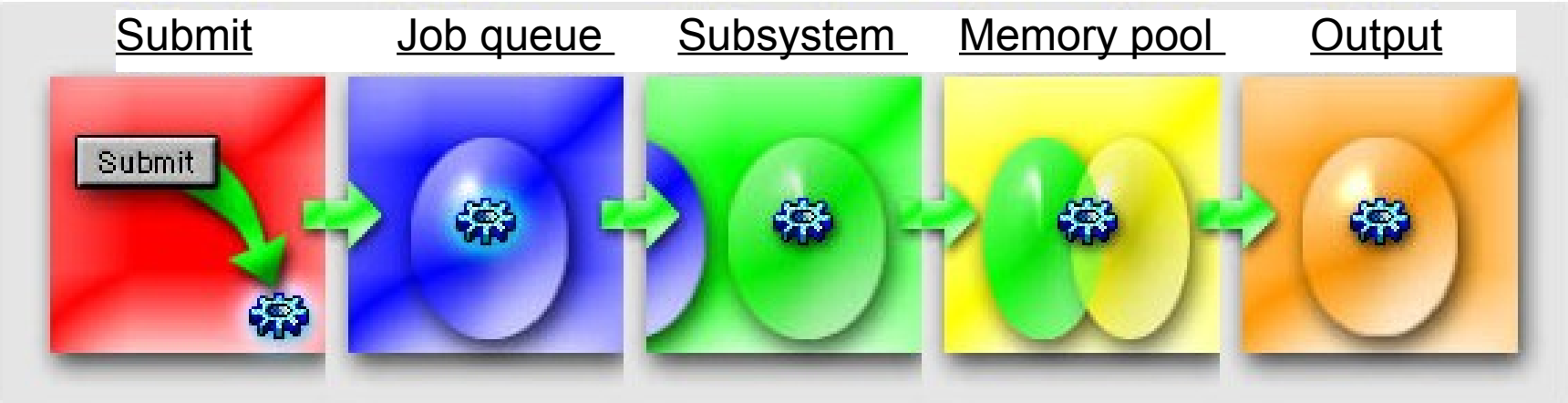
Job Types



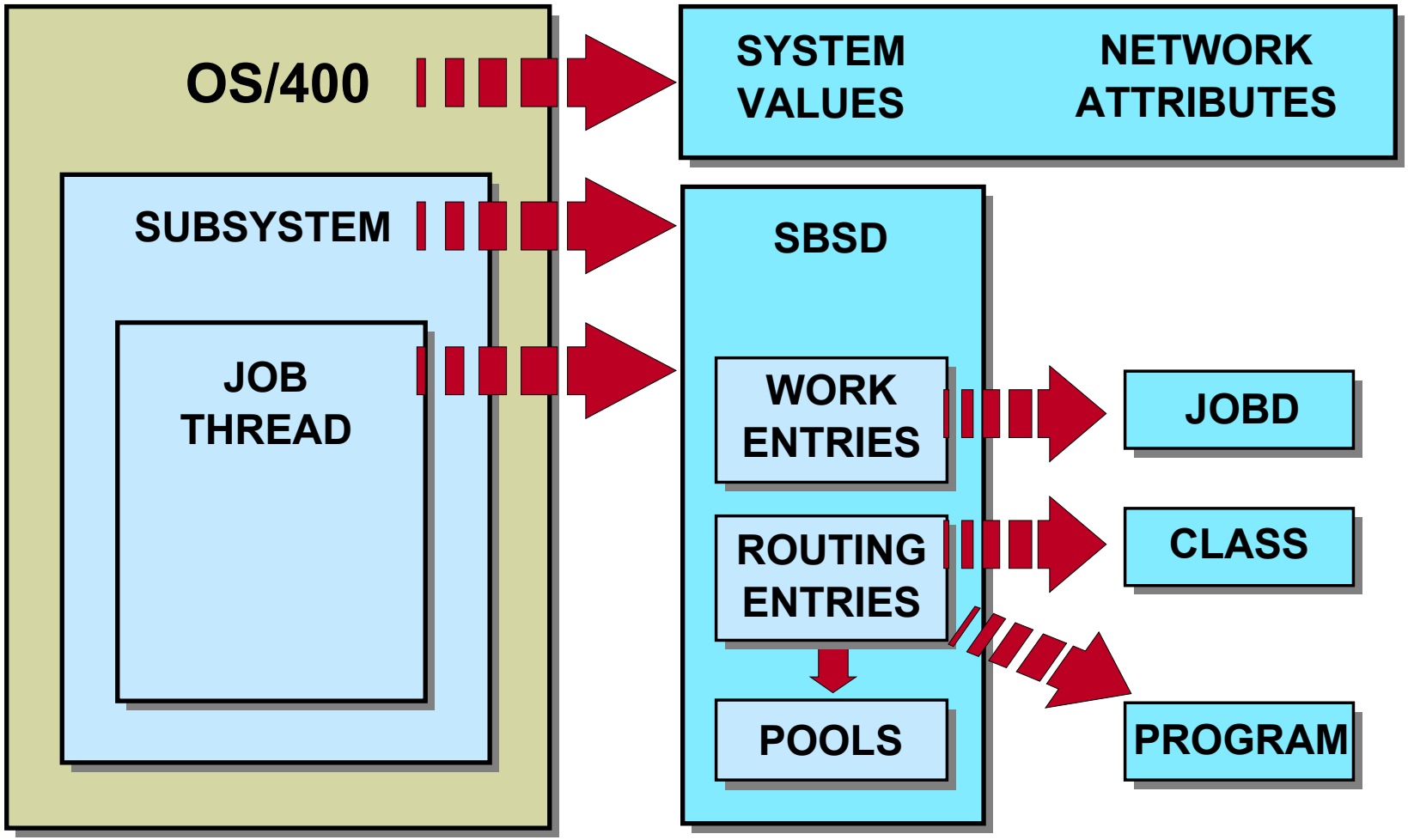
Job Properties



The Life Cycle of a Job



Work Management Objects



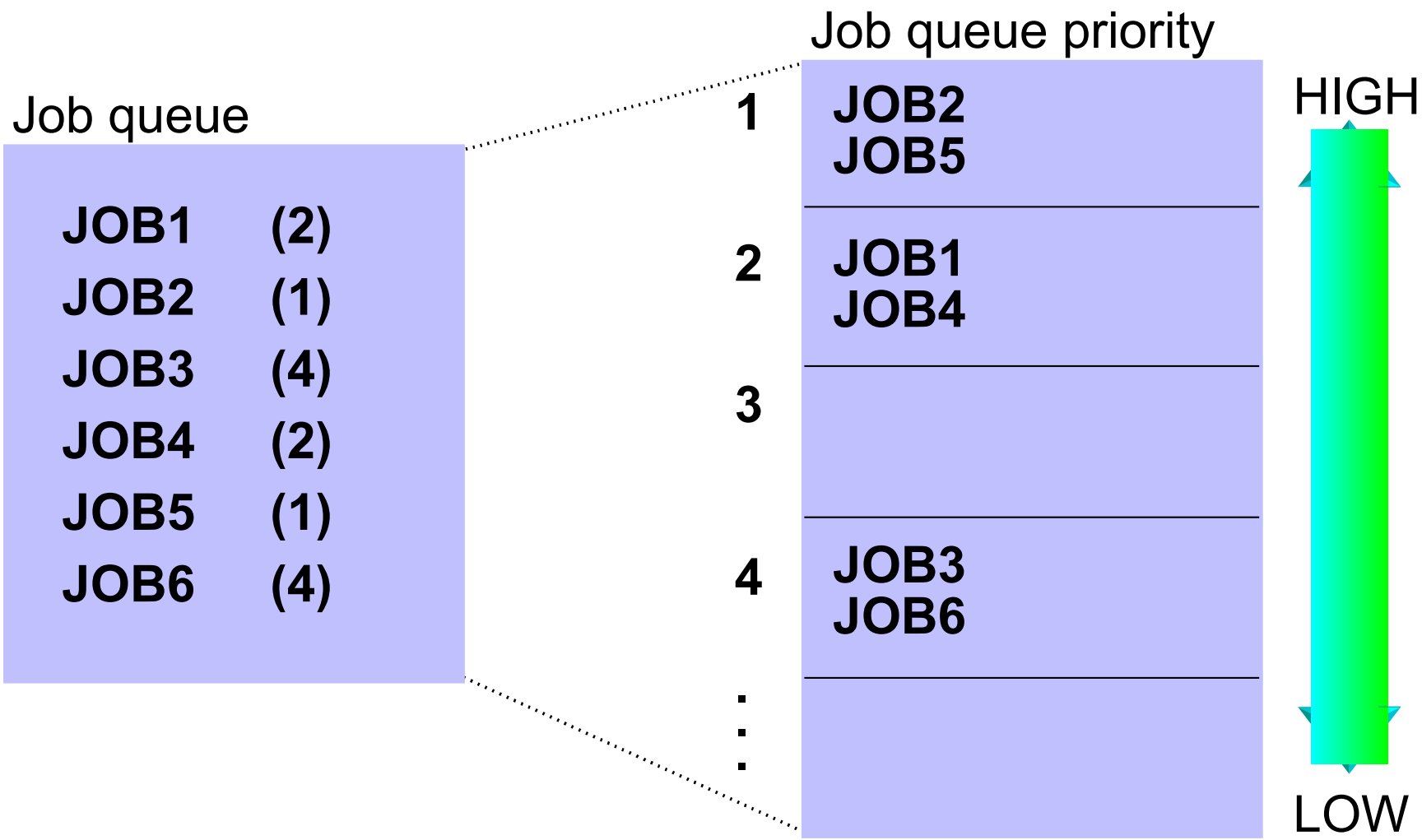
System Values

Name in iSeries Navigator	Description of system value	Name in character-based interface
Dynamically adjust job priorities of interactive jobs	Sets the job priorities of interactive jobs	QDYNPTYADJ
Dynamically adjust job priorities within priority bands	Sets job priorities within priority bands to be dynamically adjusted.	QDYNPTYSCD
Automatically adjust memory pools and activity levels	Specifies when to automatically adjust memory pools and activity levels.	QPFRADJ
Maximum eligible threads	Specifies the maximum number of eligible threads.	QMAXACTLVL
Machine memory pool size	Specifies the size of the machine memory pool.	QMCHPOOL
Base memory pool minimum size	Specifies the minimum base memory pool.	QBASPOOL
Base memory pool maximum eligible threads	Specifies maximum number of eligible threads.	QBASACTLVL
Move interactive jobs to base pool at end of time slice	Specifies whether to move interactive jobs to base pool at the end of the time slice or not.	QTSEPOOL

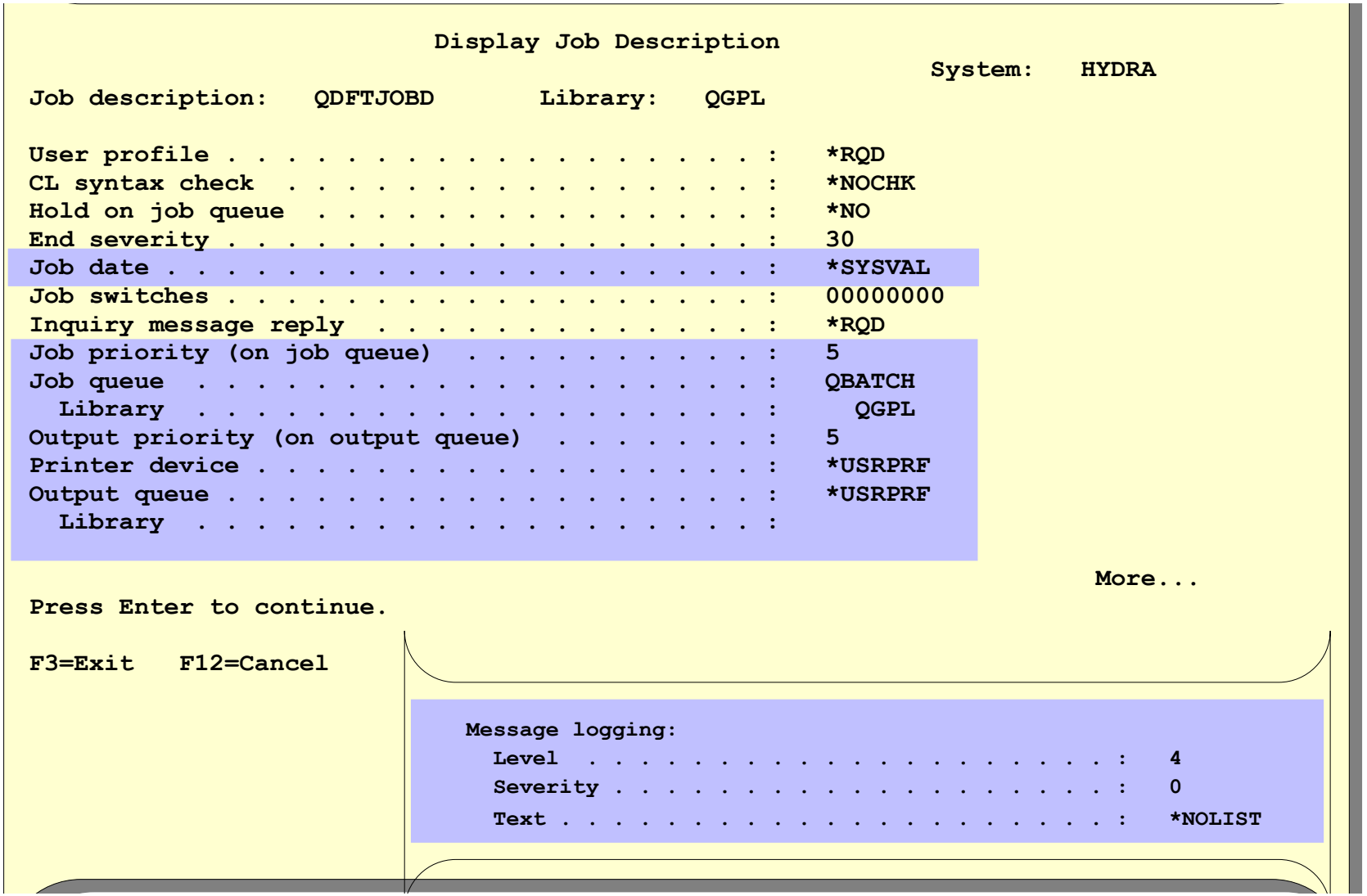
Network Attribute

Current System Name	**ABC**
Local Network ID	**APPN**
APPN Node type	***ENDNODE**
Alert Status	***OFF**
Alert Primary Focal Point	***NO**
Message Queue/Obj. Dist. **Output Queue/Obj. Dist.**	**QSYS/QSYSOPR** **QGPL/QPRINT**
Action for DDM Request	***OBJAUT**
PC Support Request Access	***OBJAUT**

Job Queue



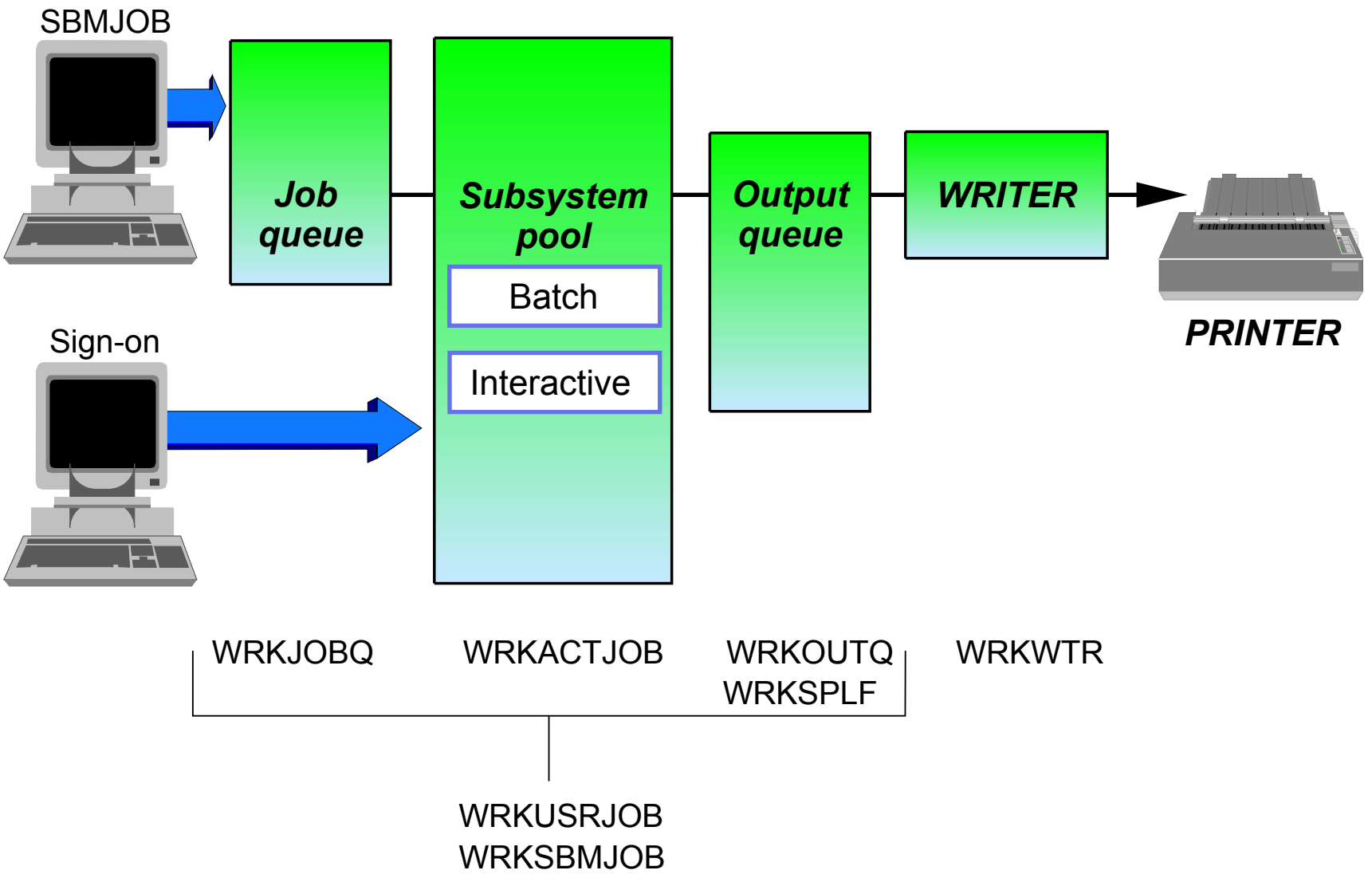
Job Description



User Profile

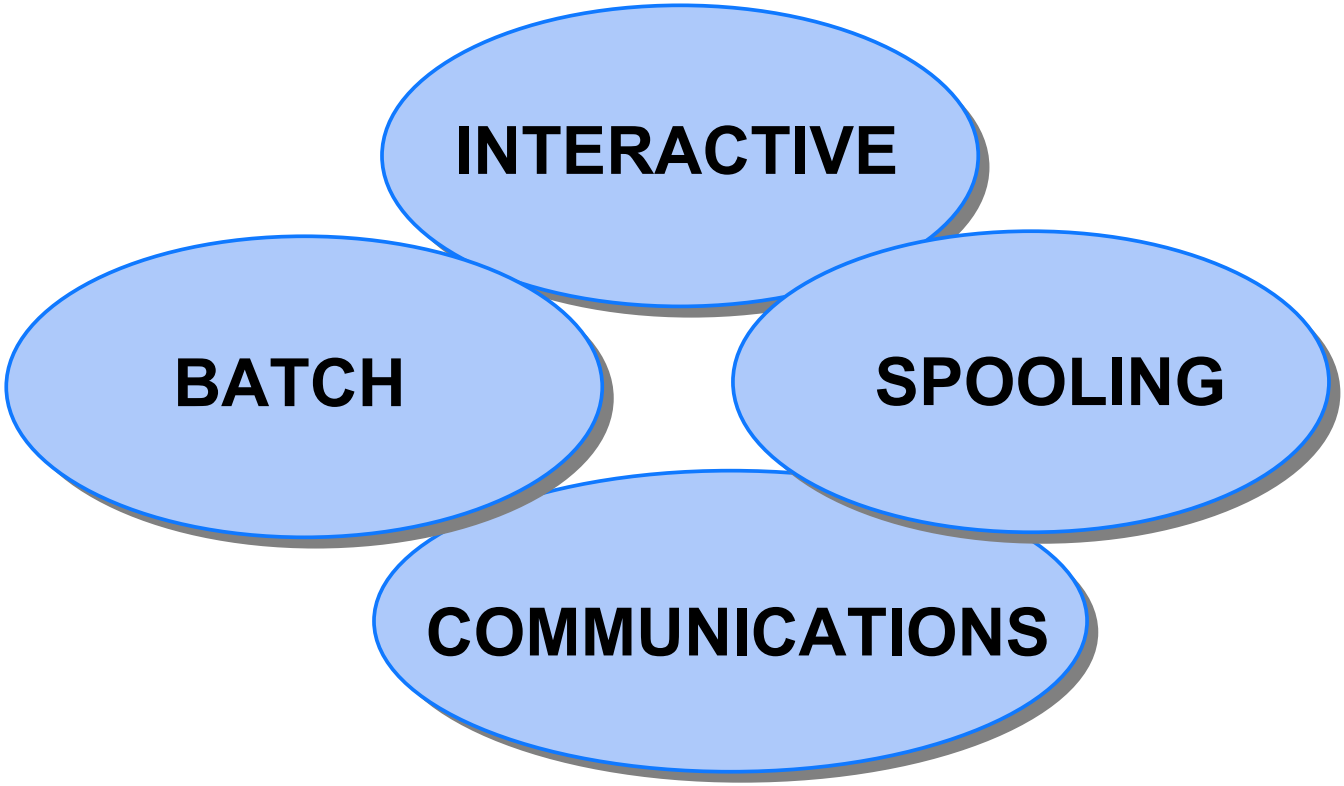
```
                    Display User Profile - Basic

User profile . . . . . . . . . . . . . . . :   AS2401

Storage information:
  Maximum storage allowed  . . . . . . . . :   *NOMAX
  Storage used . . . . . . . . . . . . . . :   312
  Storage used on independent ASP  . . . . :   *NO
Highest scheduling priority  . . . . . . . :   3
Job description  . . . . . . . . . . . . . :   QDFTJOBD
  Library  . . . . . . . . . . . . . . . . :     QGPL
Accounting code  . . . . . . . . . . . . . :
Message queue  . . . . . . . . . . . . . . :   AS2401
  Library  . . . . . . . . . . . . . . . . :     QUSRSYS
Message queue delivery . . . . . . . . . . :   *NOTIFY
Message queue severity . . . . . . . . . . :   00
Output queue . . . . . . . . . . . . . . . :   AS24OUTQ
  Library  . . . . . . . . . . . . . . . . :     AS2401
Printer device . . . . . . . . . . . . . . :   *WRKSTN
                                                                More...
Press Enter to continue.

F3=Exit   F12=Cancel
```

Job Processing



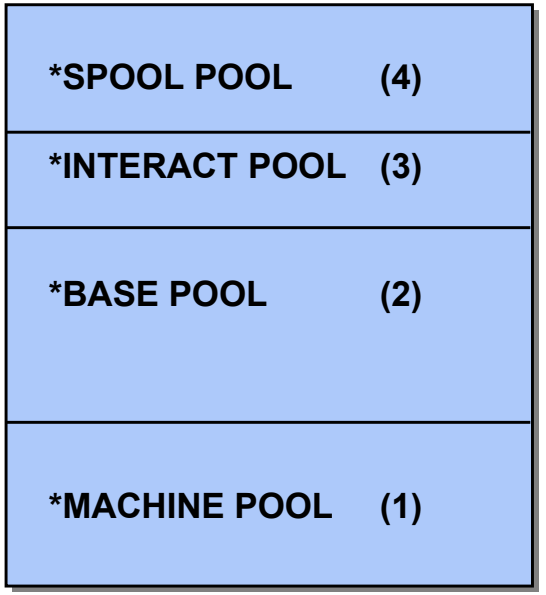
Subsystems

A specialized environment for handling a certain type of work or function, such as:



Storage Pools

- Logical Areas of Main Storage for Jobs:
  - POOL SIZE sets amounts of storage
  - Noncontiguous pages
  - Deallocated when set to *NOSTG (No storage)
  - PAGING is restricted to pool
  - ACTIVITY LEVEL determines how many job threads may have main storage
  - Two types of pools: SHARED AND PRIVATE



Subsystem Description (1/4)

```
                        Display Subsystem Description
Subsystem description:  QINTER                              System: ABC
Status:  ACTIVE                                  Library:  QSYS

Select one of the following:
 1. Operational attributes
 2. Pool definitions
 3. Autostart job entries
 4. Work station name entries
 5. Work station type entries
 6. Job queue entries
 7. Routing entries
 8. Communications entries
 9. Remote location name entries
10. Prestart job entries

30. All of the above

Selection or command
==>_____________________________________
```

Operational Attributes

Maximum jobs in subsystem	*NOMAX
Sign-on display file	QSYS/QDSIGNON

Pool Definitions

Pool ID	Storage Size (KB)
1	*BASE
2	*INTERACT

Subsystem Description (2/4)

```
                        Display Subsystem Description
Subsystem description:  QINTER                          System: ABC
Status:  ACTIVE                                 Library:  QSYS

Select one of the following:
 1. Operational attributes
 2. Pool definitions
 3. Autostart job entries
 4. Work station name entries
 5. Work station type entries
 6. Job queue entries
 7. Routing entries
 8. Communications entries
 9. Remote location name entries
10. Prestart job entries

30. All of the above

Selection or command
==>_____________________________
```

WORKSTATION Entries

Type	Jobd
*ALL	*USRPRF

Name	Jobd
DSP03	*USRPRF

Job Queue Entries

Jobq
QBATCH

PRESTART Job Entries

Prog	Jobd
PROG01	JOBDAPPC

AUTOSTART Job Entries

Job	Jobd
START	STARTJD

Communication Entries

Dev	Jobd
*ALL	*USRPRF

Subsystem Description (3/4)

```
                        Display Subsystem Description
Subsystem description:  QINTER                          System: ABC
Status:  ACTIVE                                 Library:  QSYS

Select one of the following:
 1. Operational attributes
 2. Pool definitions
 3. Autostart job entries
 4. Work station name entries
 5. Work station type entries
 6. Job queue entries
 7. Routing entries
 8. Communications entries
 9. Remote location name entries
10. Prestart job entries

30. All of the above

Selection or command
==>_____________________________________________________________
```

Routing Entries

SEQNBR	CMPVAL	CLASS	PGM	POOL ID
10	QCMDB	QBATCH	QCMD	1
50	QCMDI	QINTER	QCMD	2
9999	*ANY	QBATCH	QCMD	1

Class Information

	QBATCH	QINTER
Run Priority	50	20
Time Slice	5000 ms	2000 ms
Purge	*NO	*YES
Wait Time	120 sec	30 sec

Subsystem Description (4/4)

```
                        Display Subsystem Description
Subsystem description:  QINTER                          System: ABC
Status:  ACTIVE                                Library:  QSYS

Select one of the following:
 1. Operational attributes
 2. Pool definitions
 3. Autostart job entries
 4. Work station name entries
 5. Work station type entries
 6. Job queue entries
 7. Routing entries
 8. Communications entries
 9. Remote location name entries
10. Prestart job entries

30. All of the above

Selection or command
==>_____________________________
```

Job queue entries in subsystem description

JOBQ	JOBQA	JOBQB	JOBQC
SEQNBR	1O	20	30
MAXACT	*NOMAX	1	1
MAXPTY1 . MAXPTY9	*NOMAX : :	*NOMAX : :	*NOMAX :
	JOB 2 JOB 1	JOB 4 JOB 3	JOB 5

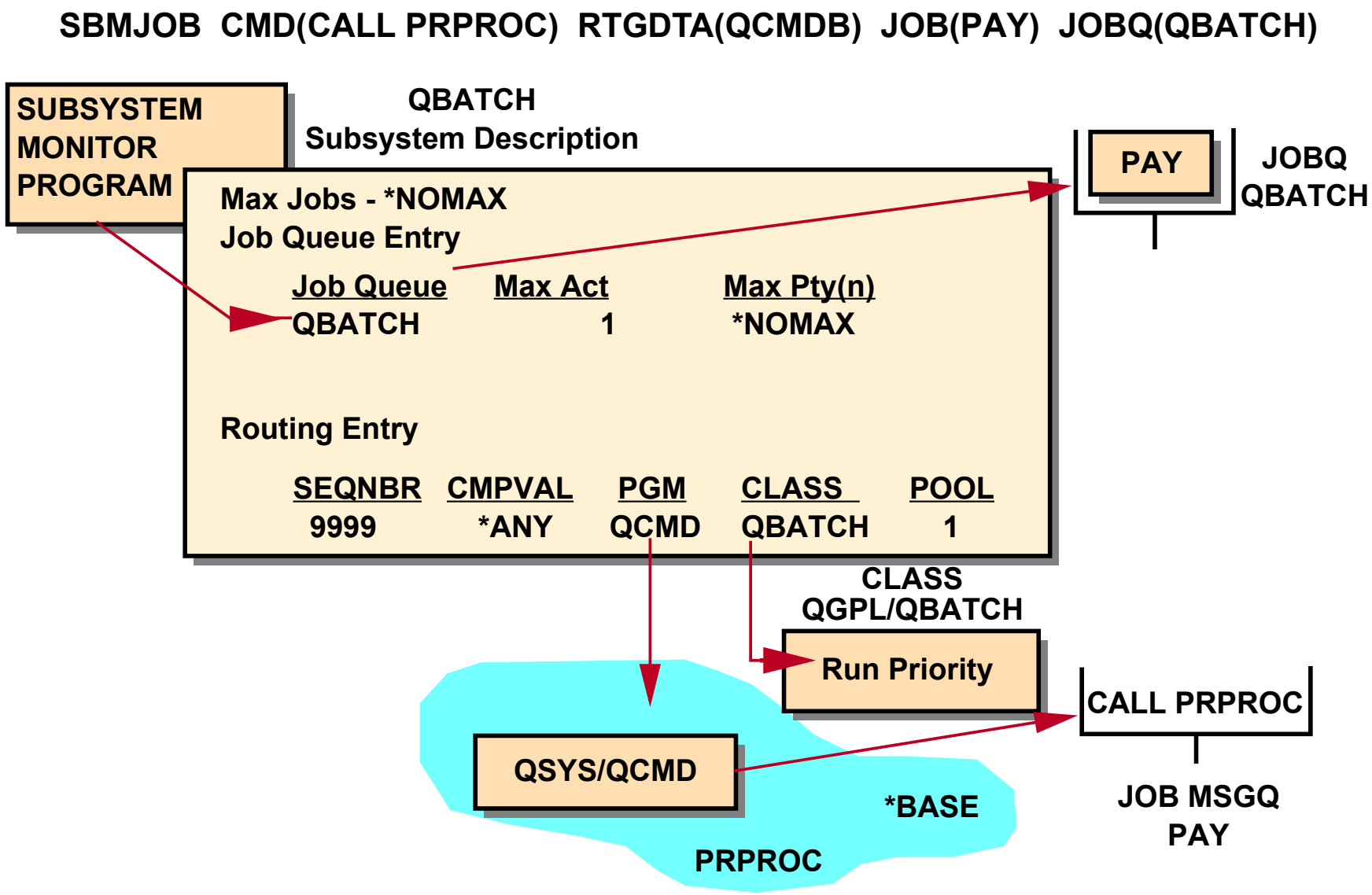
Jobs 1 to 5 have equal priority.

Which jobs run concurrently if the subsystem description has:

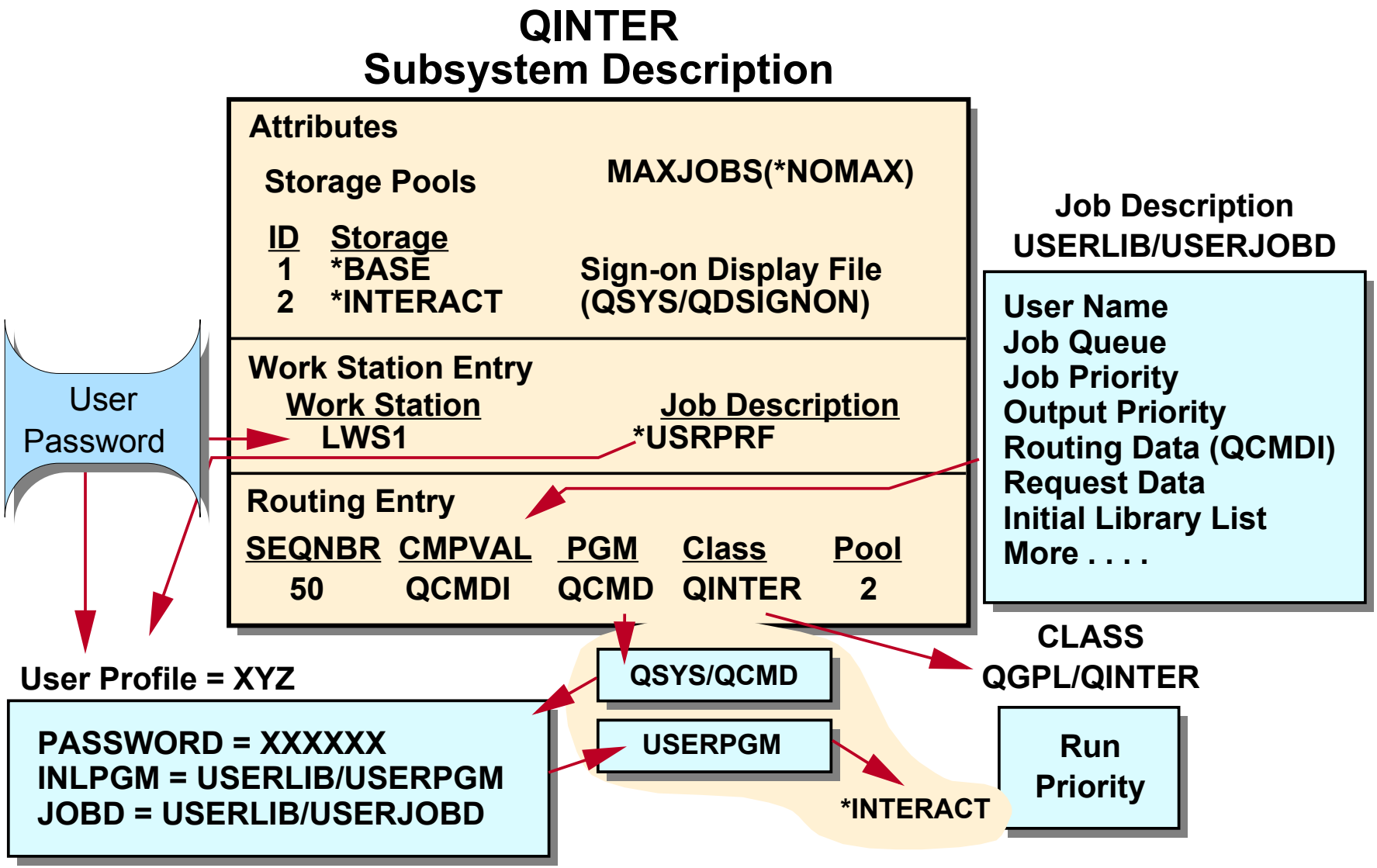
?

MAXJOBS = 2
MAXJOBS = 3
MAXJOBS = 4
MAXJOBS = *NOMAX

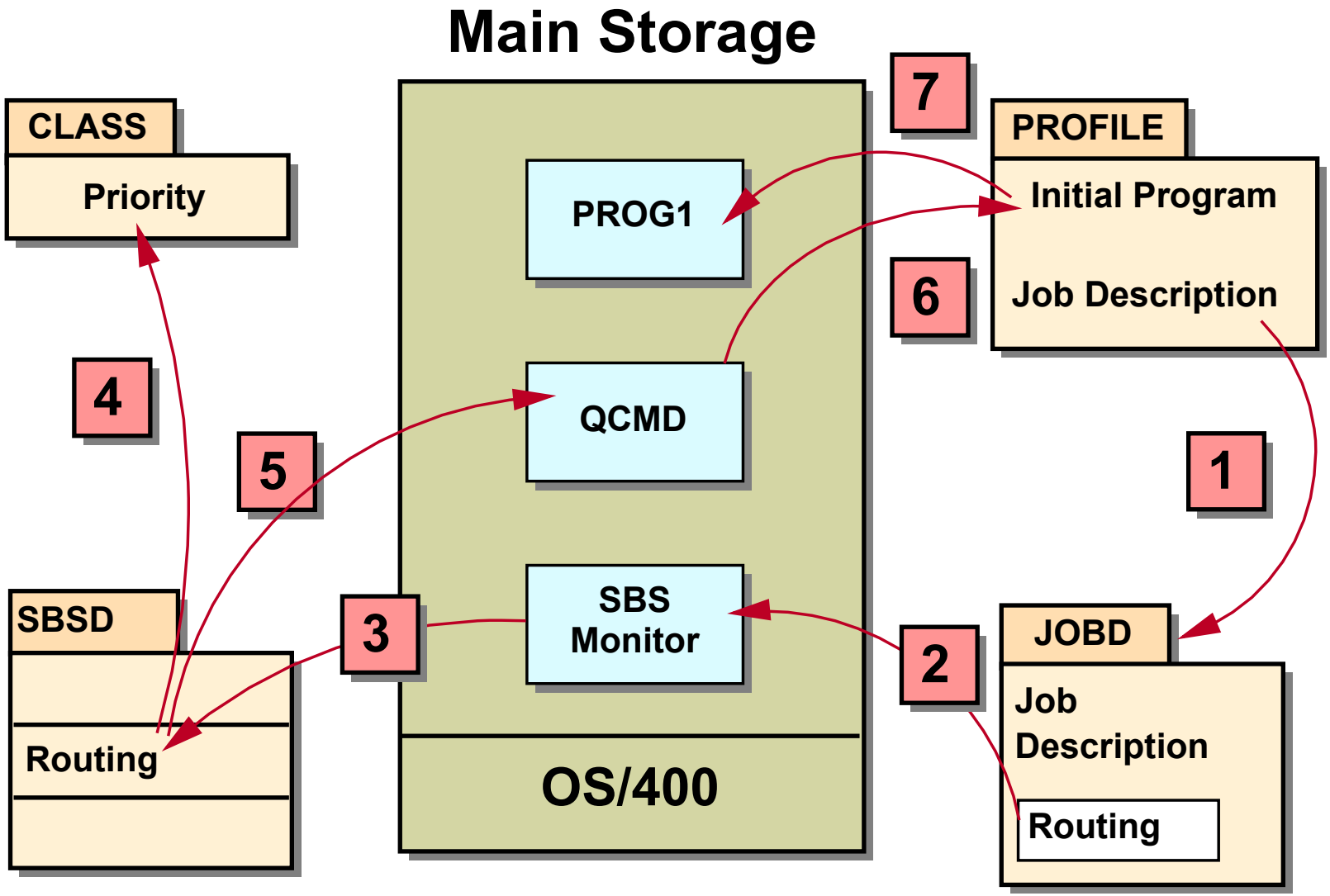
Batch Job Routing



Interactive Job Routing



Operator‘s View



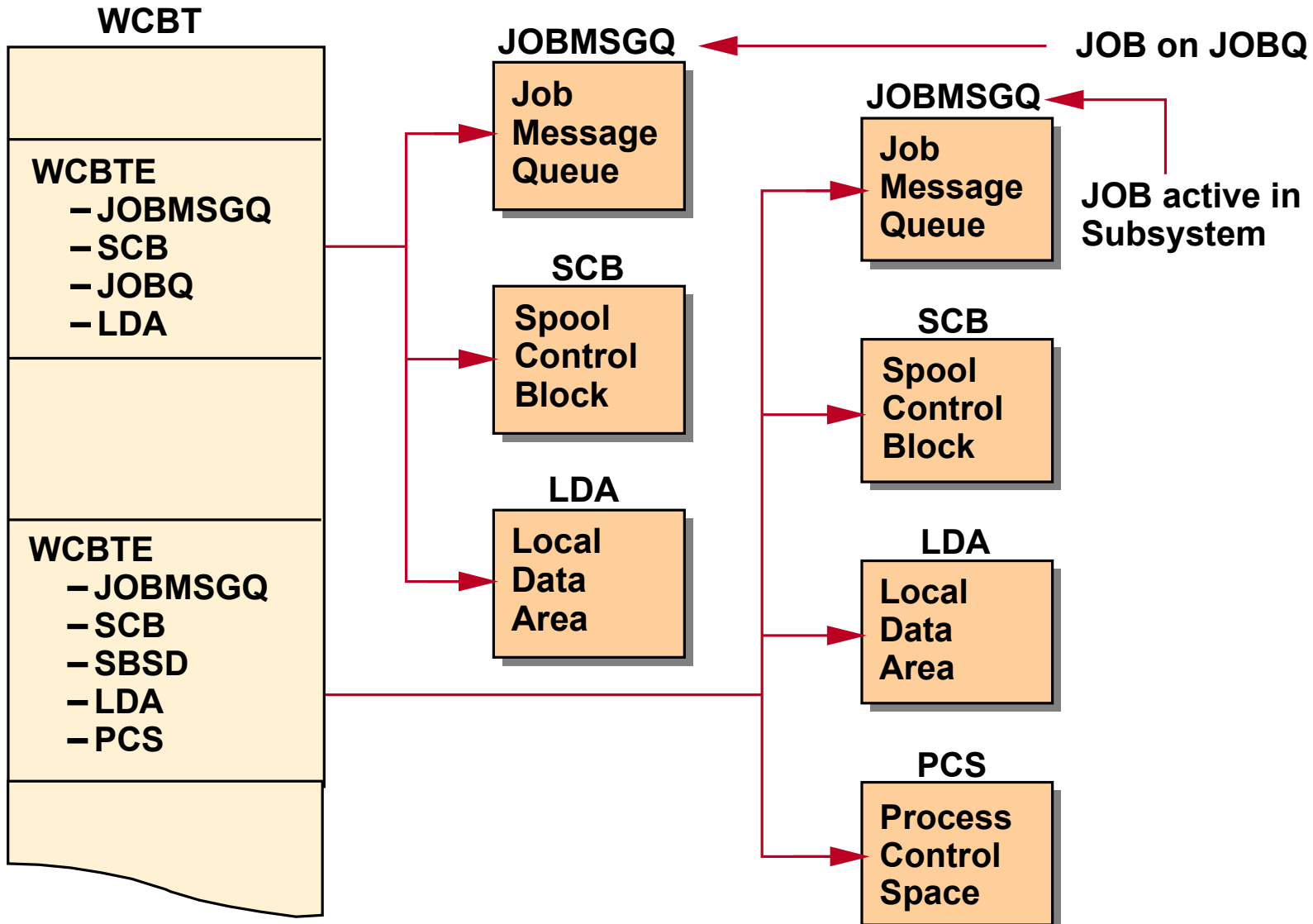
5.2 System's View

Job Structure and Task Dispatching

Three-level Hierachy of Work

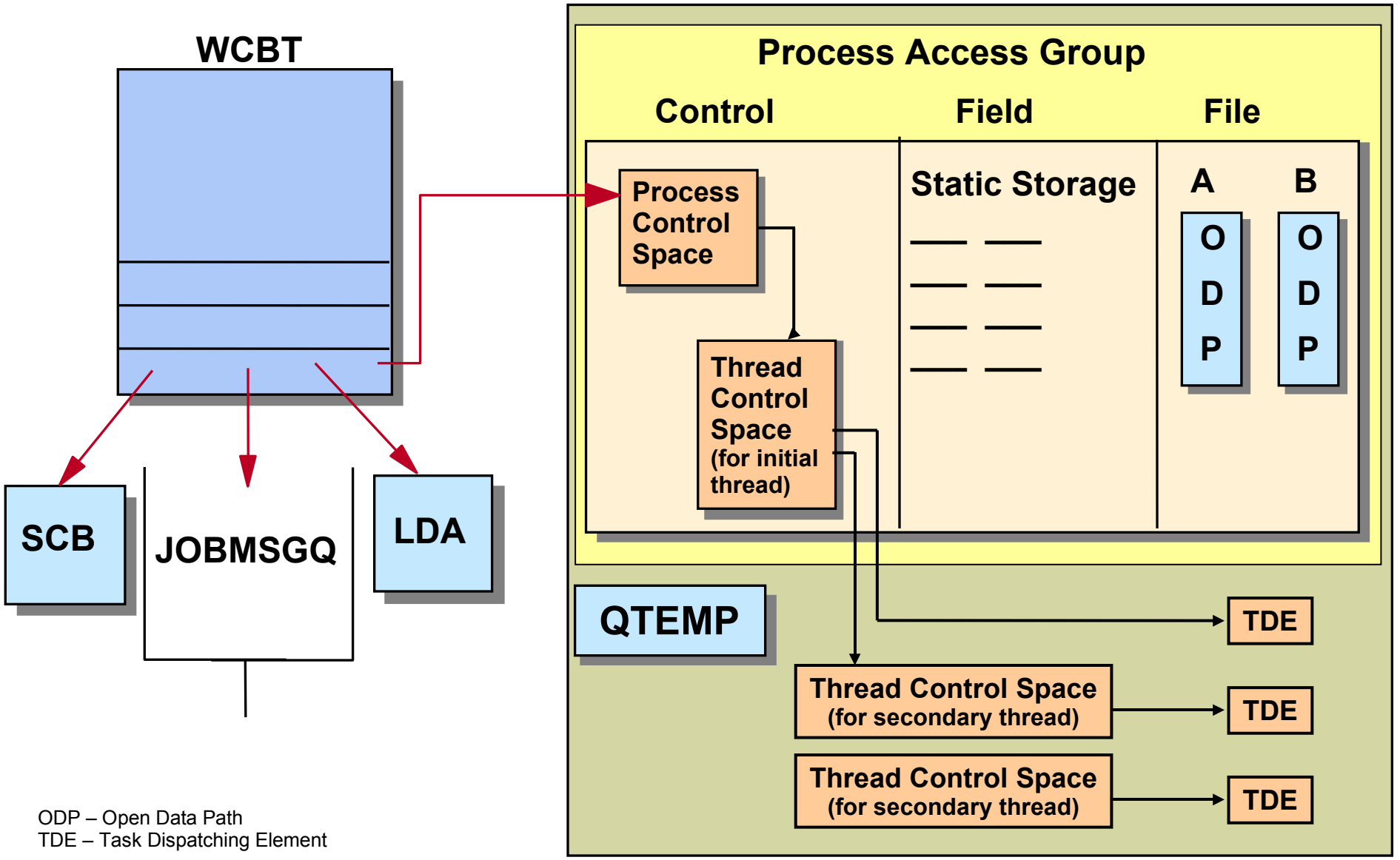
- **Process**
  - MI System object called process-control space (PCS)
  - Tie together the resources for program invocation
  - Consists of
    - A executable program (can be shared)
    - Program‘s data
    - State informations
  - Could also be defined as a program in execution
- **Job**
  - OS/400 object that contains a process structure to manage system resources
- **Thread**
  - A portion of a process
  - A process may have more than one thread
  - Each thread consists of
    - Set of registers representing the state of the processor
    - A control stack
    - Automatic program storage (local variables)
  - System resources allocated to the process are shared with other threads within the process

Permanent Job Structure Objects



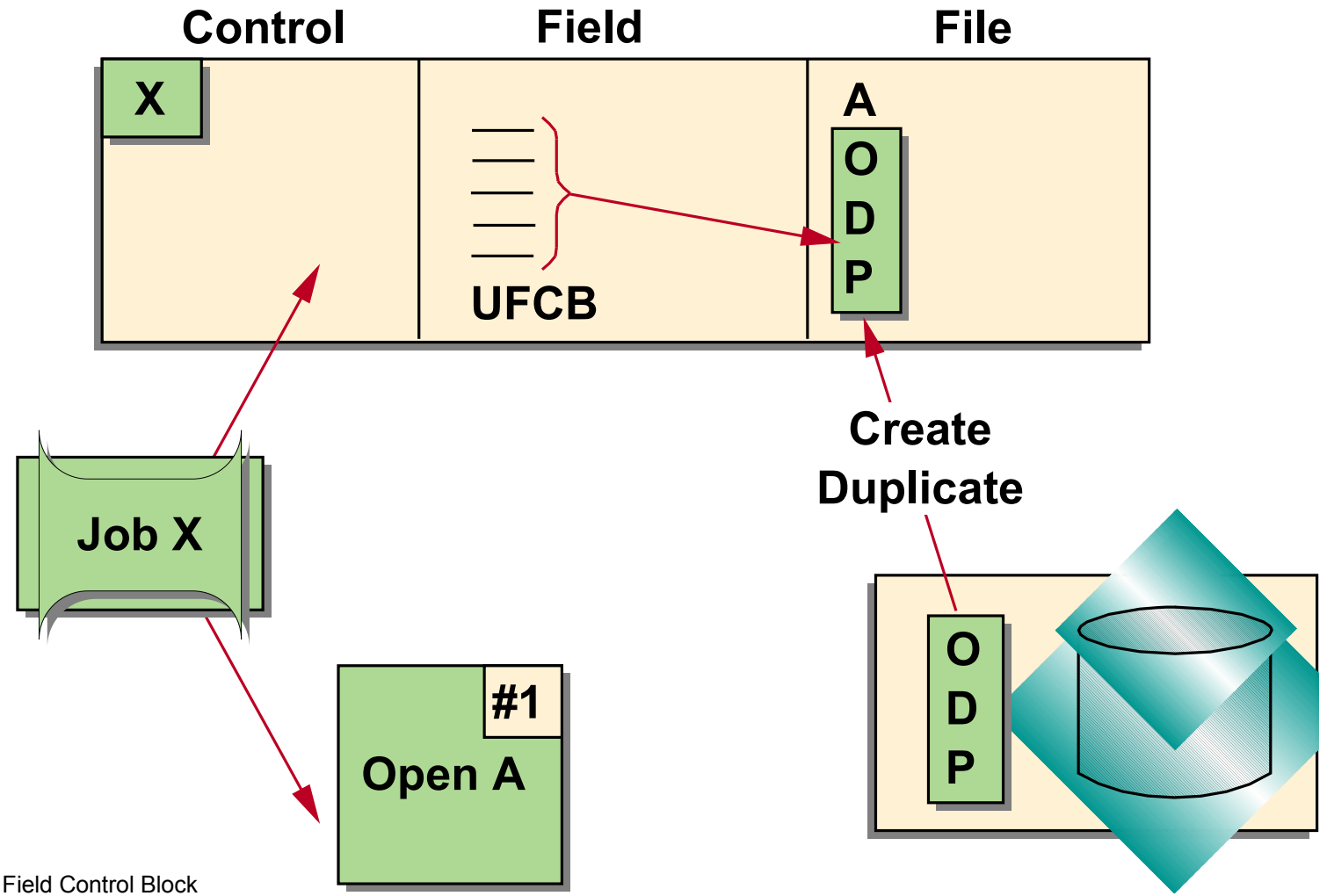
WCBT – Work Control Block Table
WCBTE – Work Control block Table Entry

Temporary Job Structure Objects



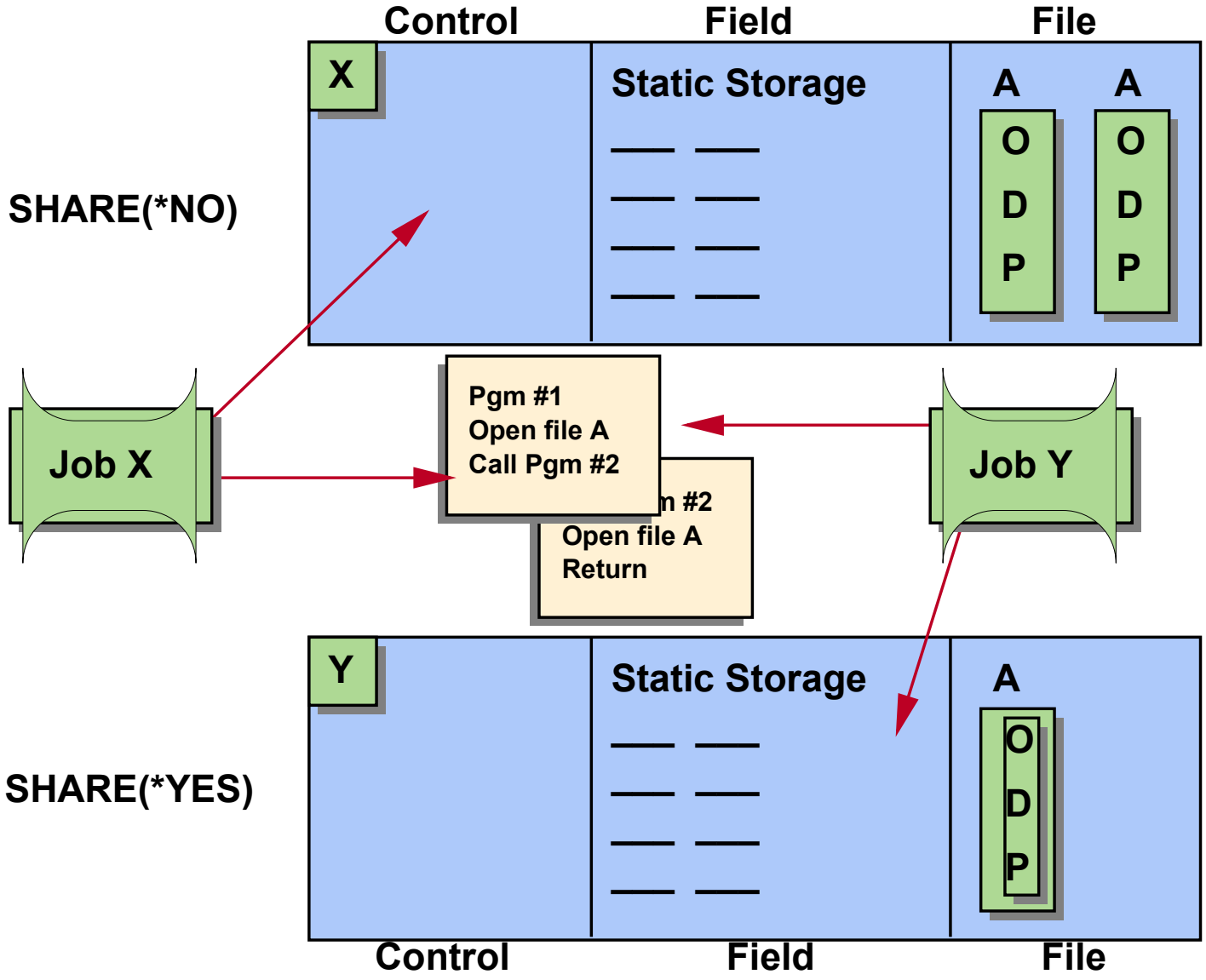
ODP – Open Data Path
TDE – Task Dispatching Element

Open Data Path (ODP)



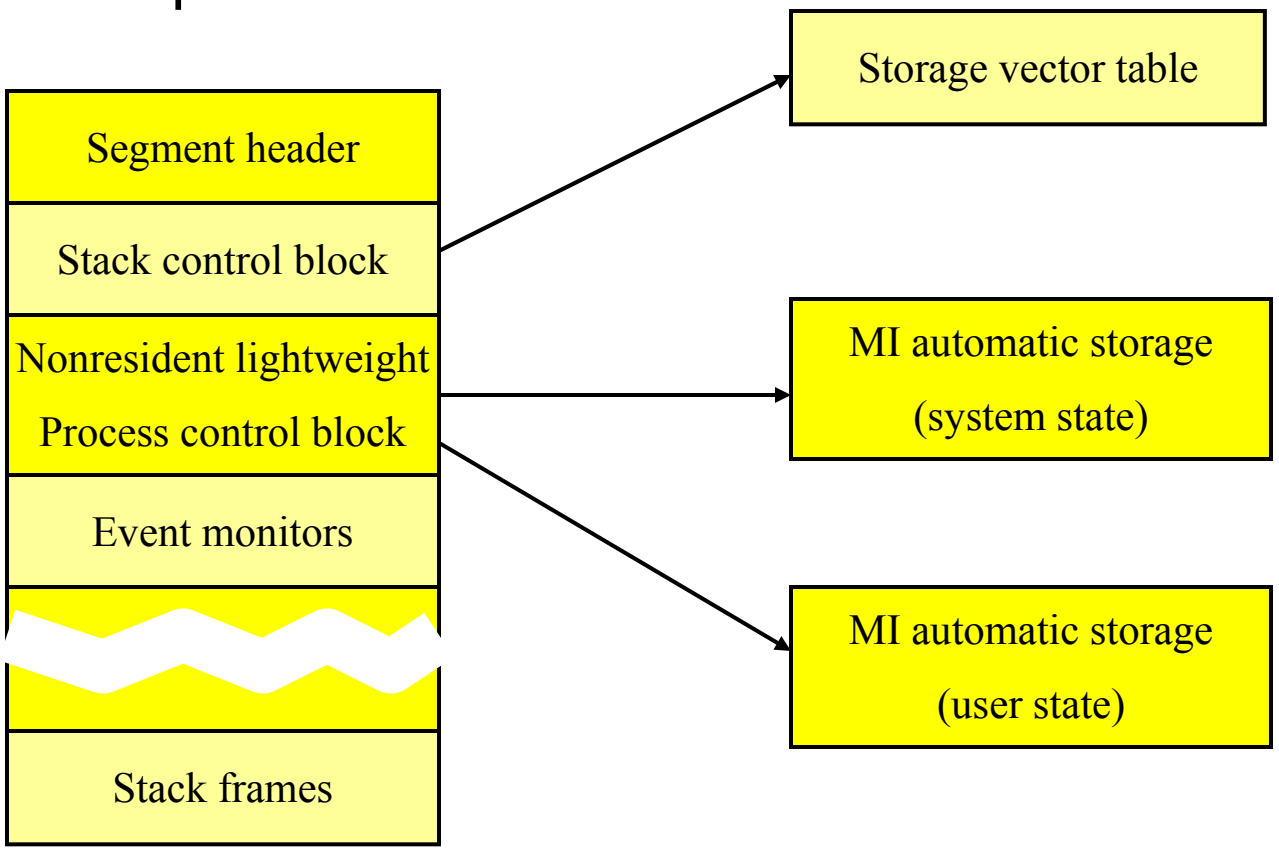
UFCB – User Field Control Block

Open Data Path Sharing



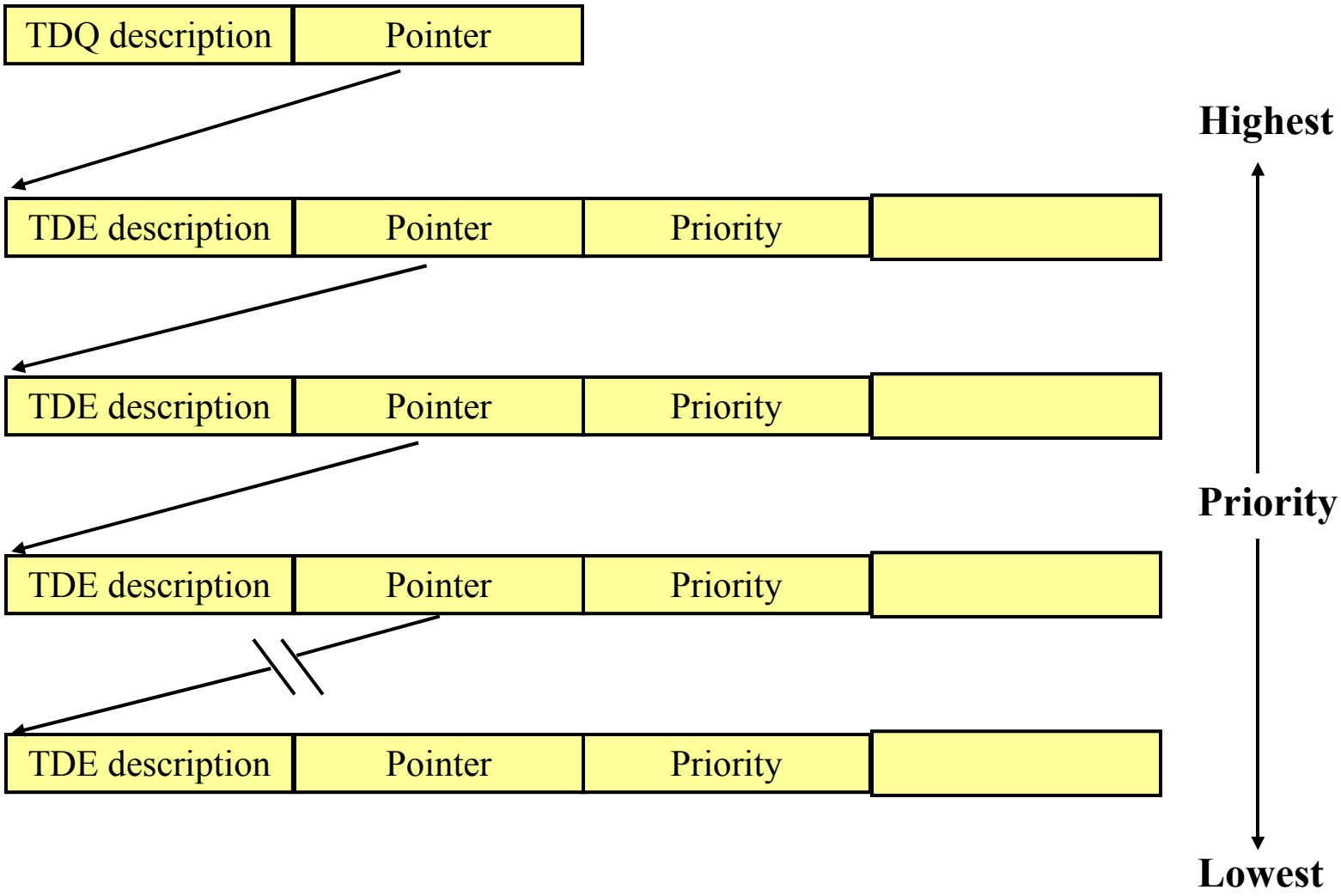
Thread-control Space (TCS)

- are temporary SLIC objects
- will be recycled at process level or at systems level when process ends



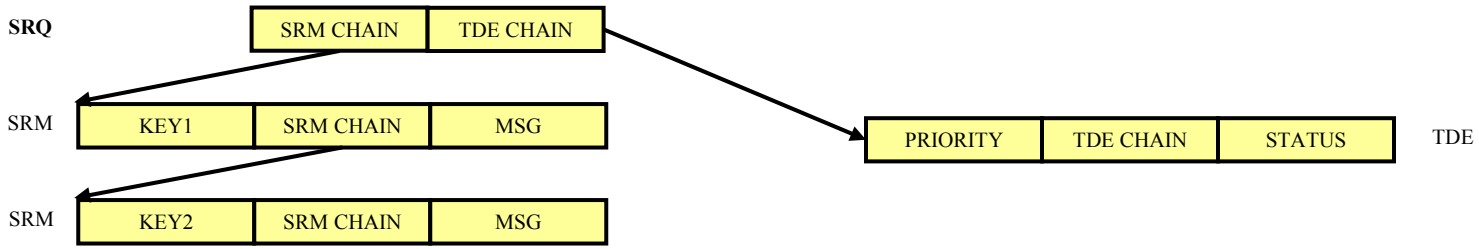
Source: Frank G. Soltis, Fortress Rochester, Fig. 13.5

Task Dispatching Queues



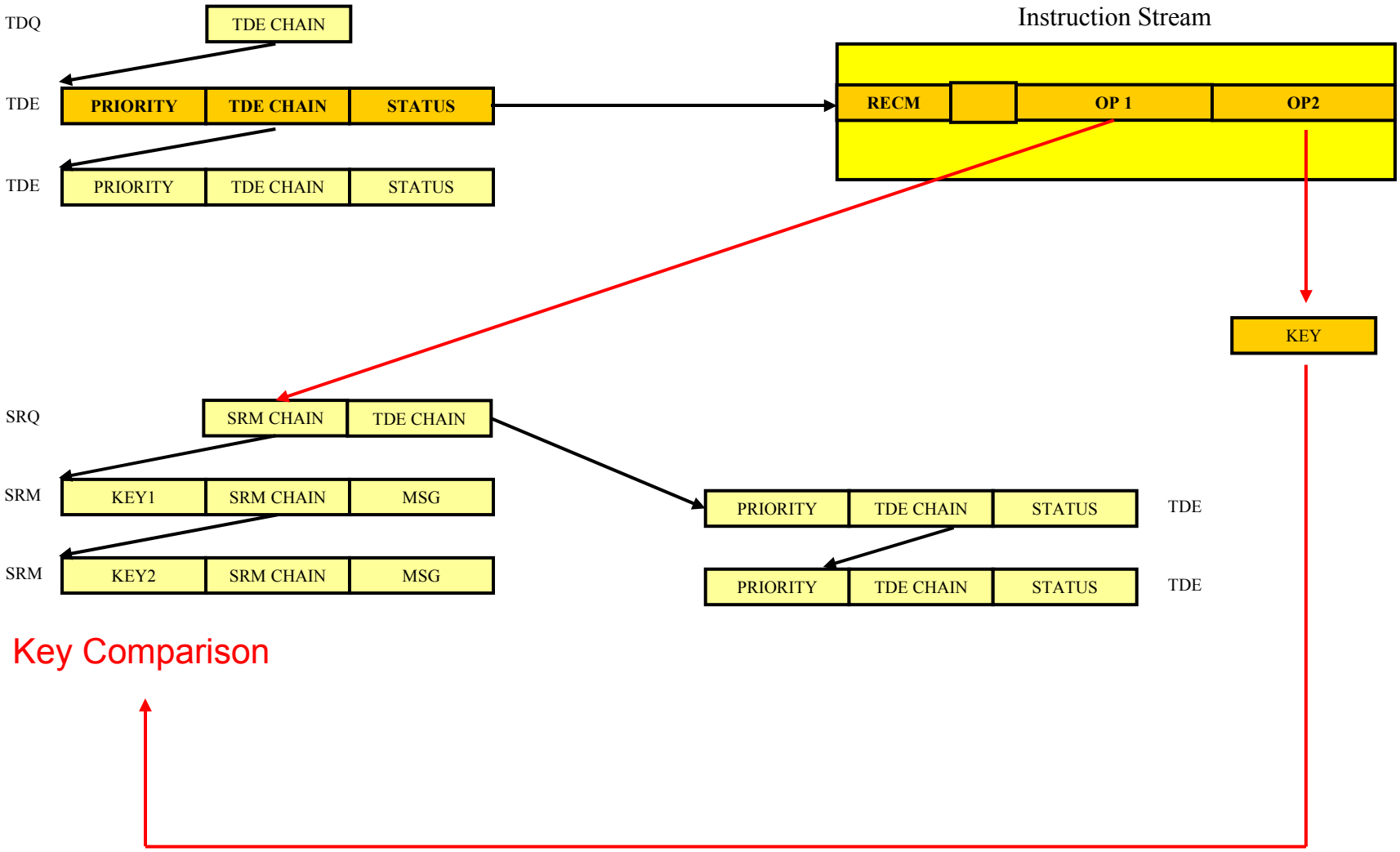
Send/Receive Queues

- Data structure used as “mailbox” for tasks
- SEND MESSAGE enqueued a SRM
- Execute a RECEIVE MESSAGE operations to obtain a message (dequeing)
  - If no message is available the task has to wait

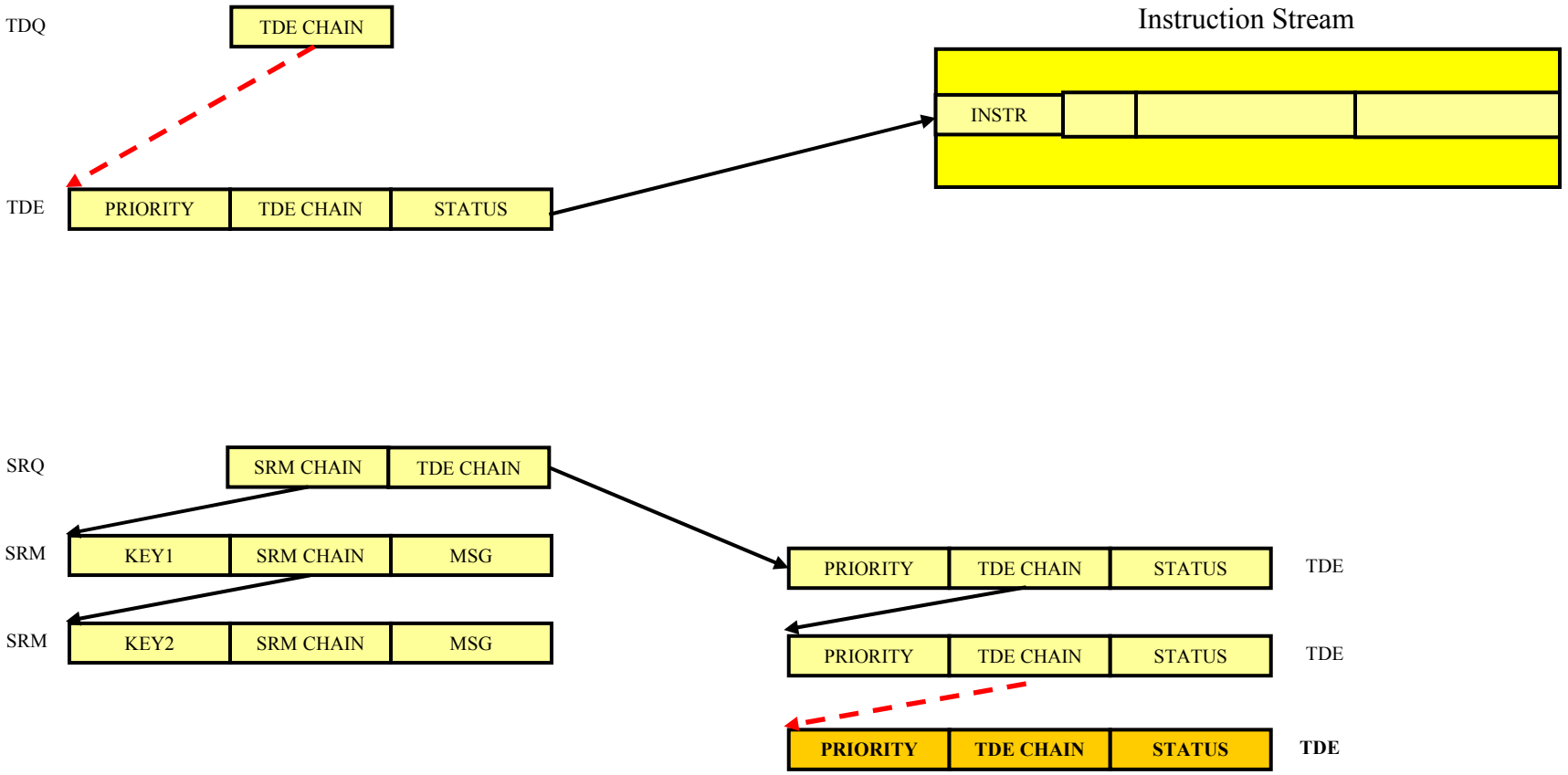


SRQ – Send/Receive Queue
SRM – Send/Receive Message
RECM – Recieve Message
SENDM – Send Message

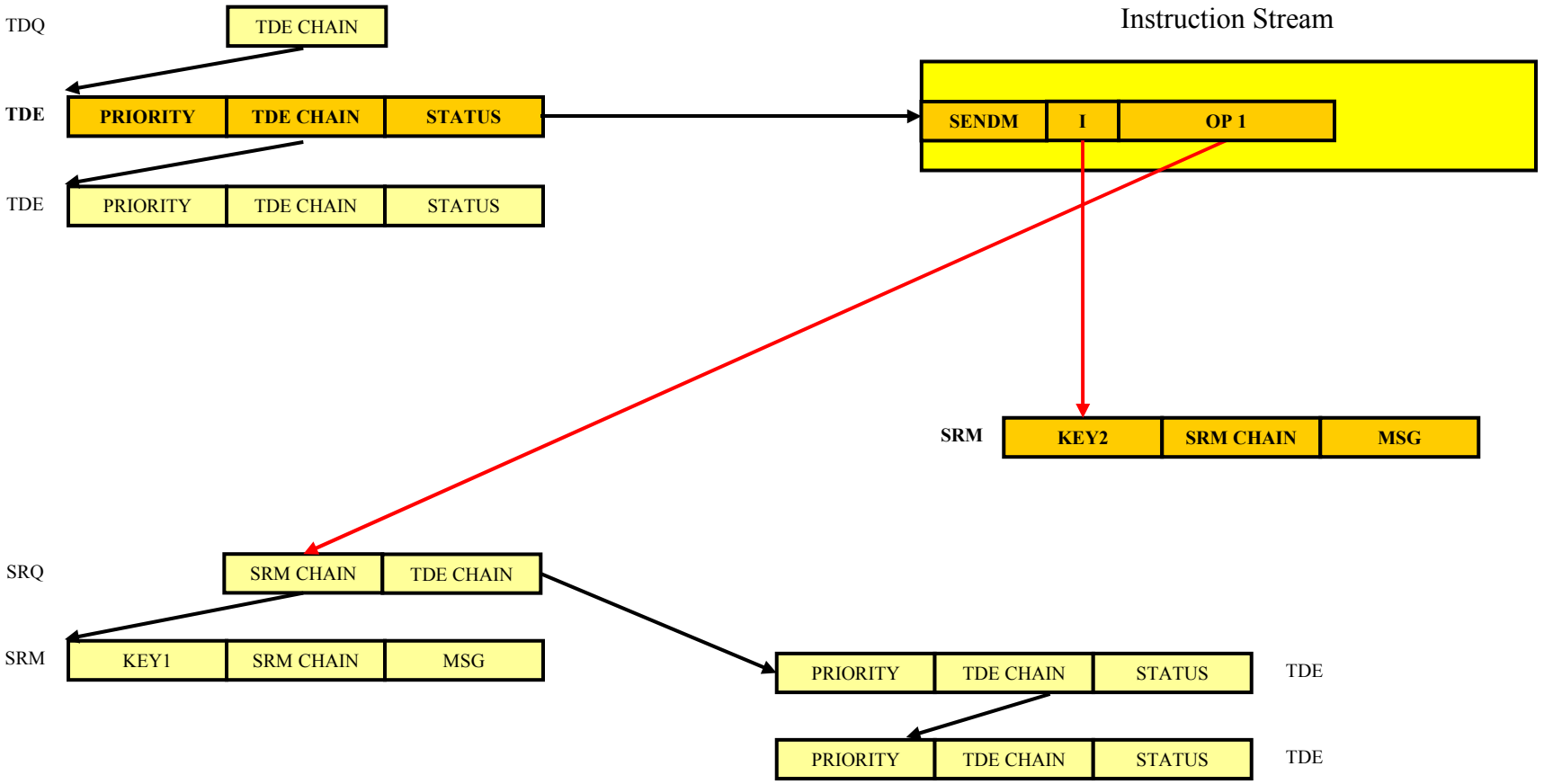
RECM



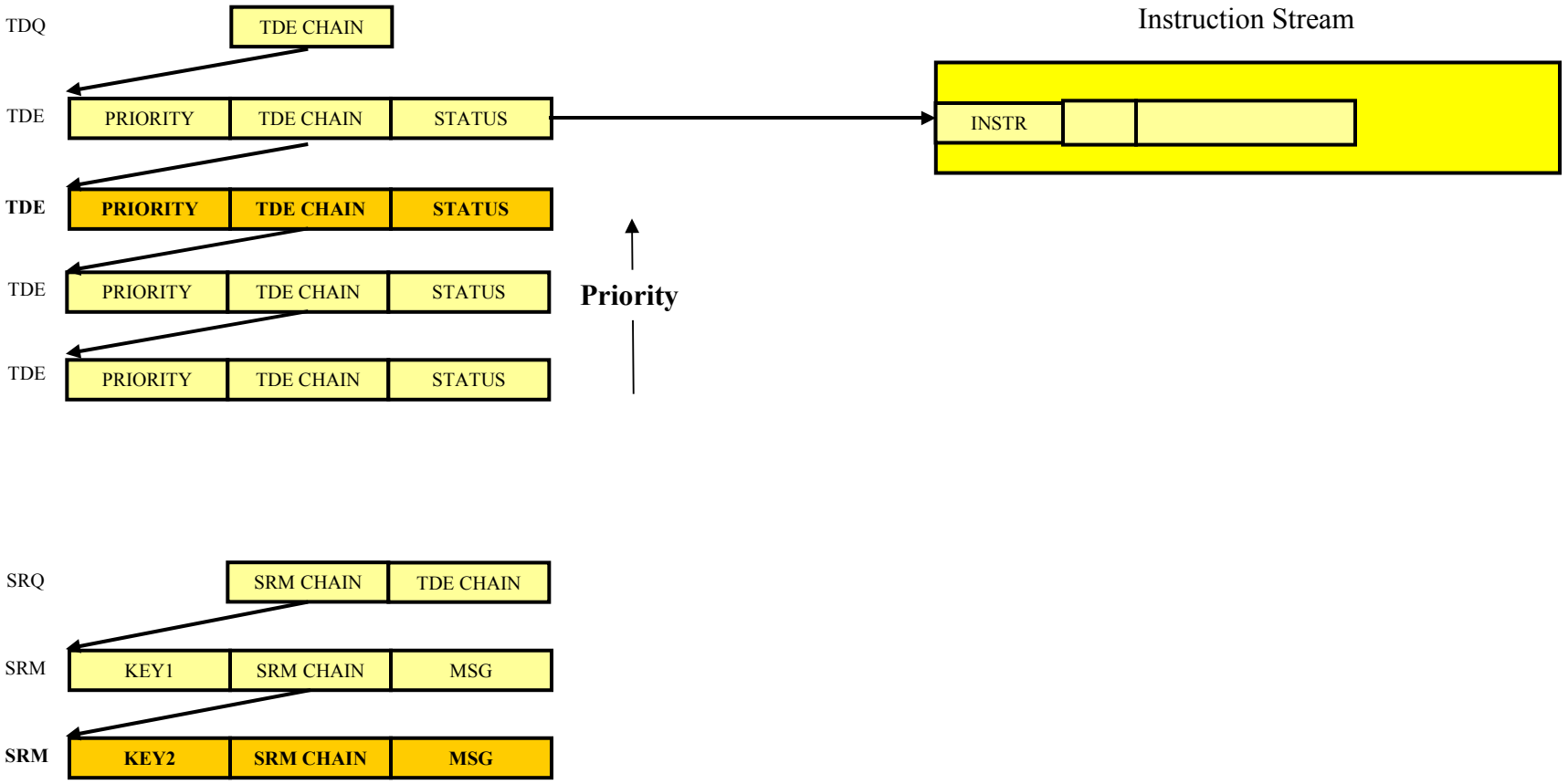
If RECM Is Unsatisfied



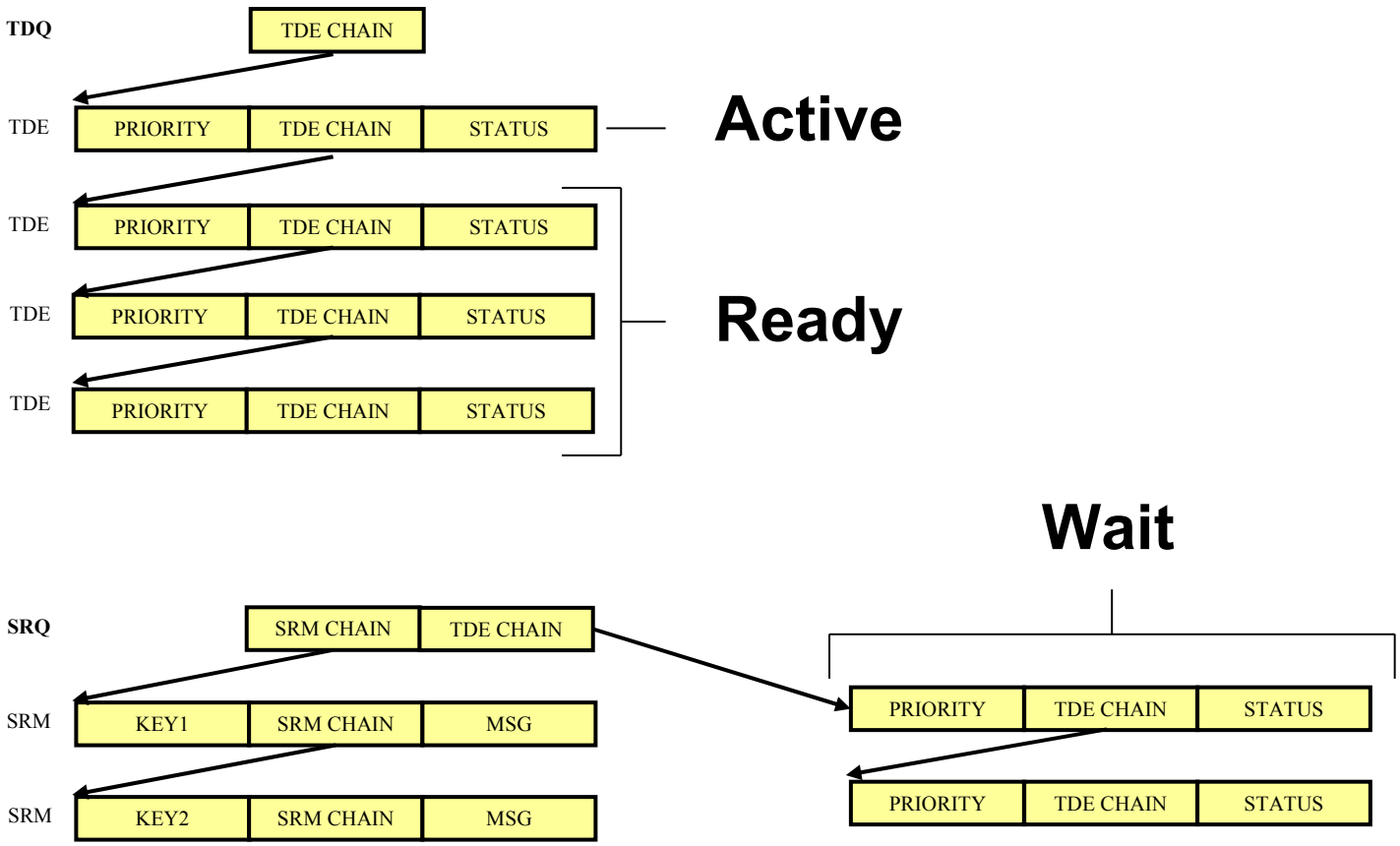
SENDM (1/2)



SENDM (2/2)

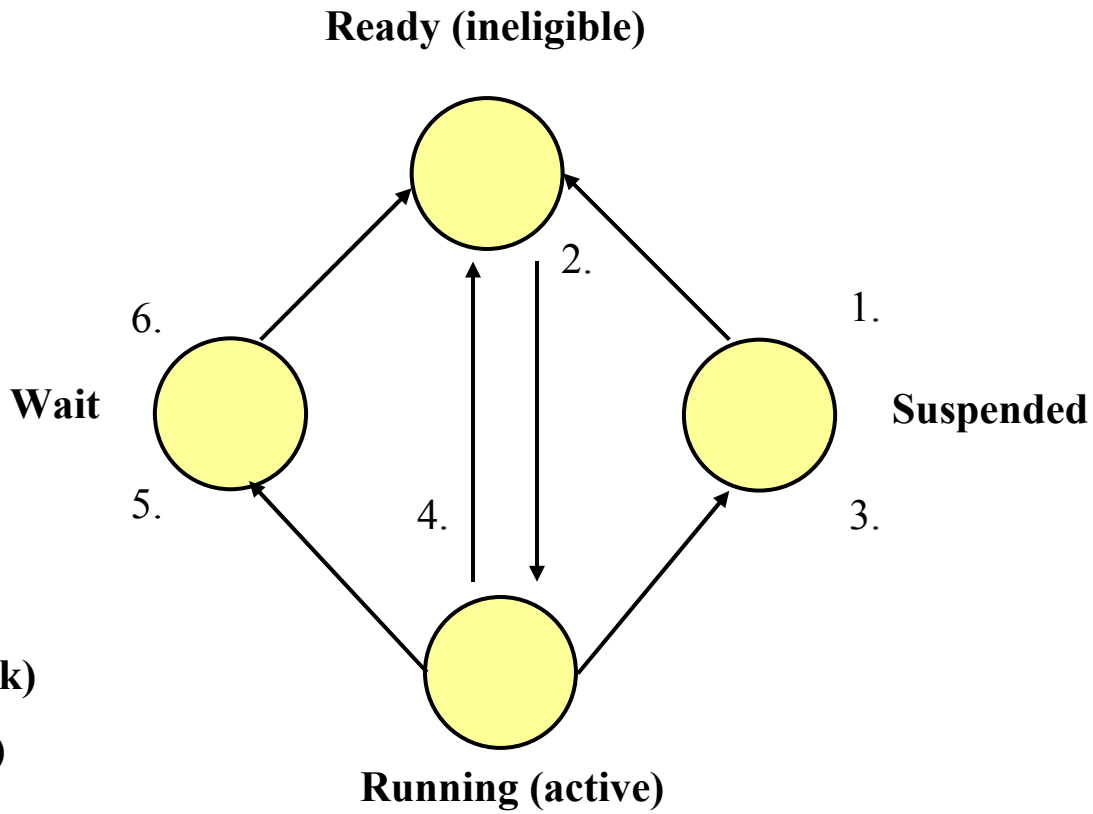


Task States



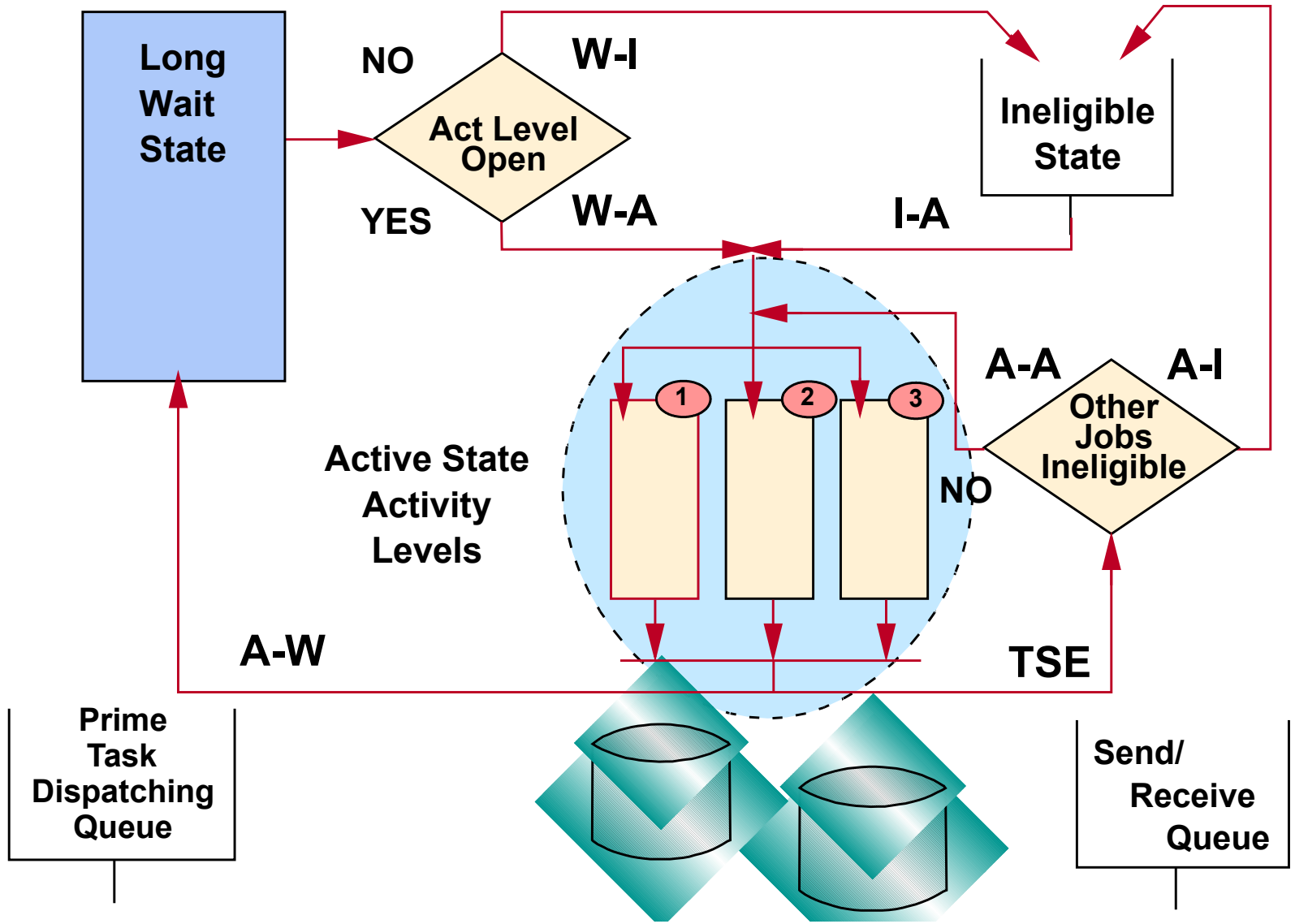
States of Tasks

1. **Initiate task (start work)**
2. **Run task (dispatch task)**
3. **Suspend task (work done)**



1. **Preempt task (by another task)**
2. **Wait(for I/O or another task)**
3. **Signal (task is done waiting)**

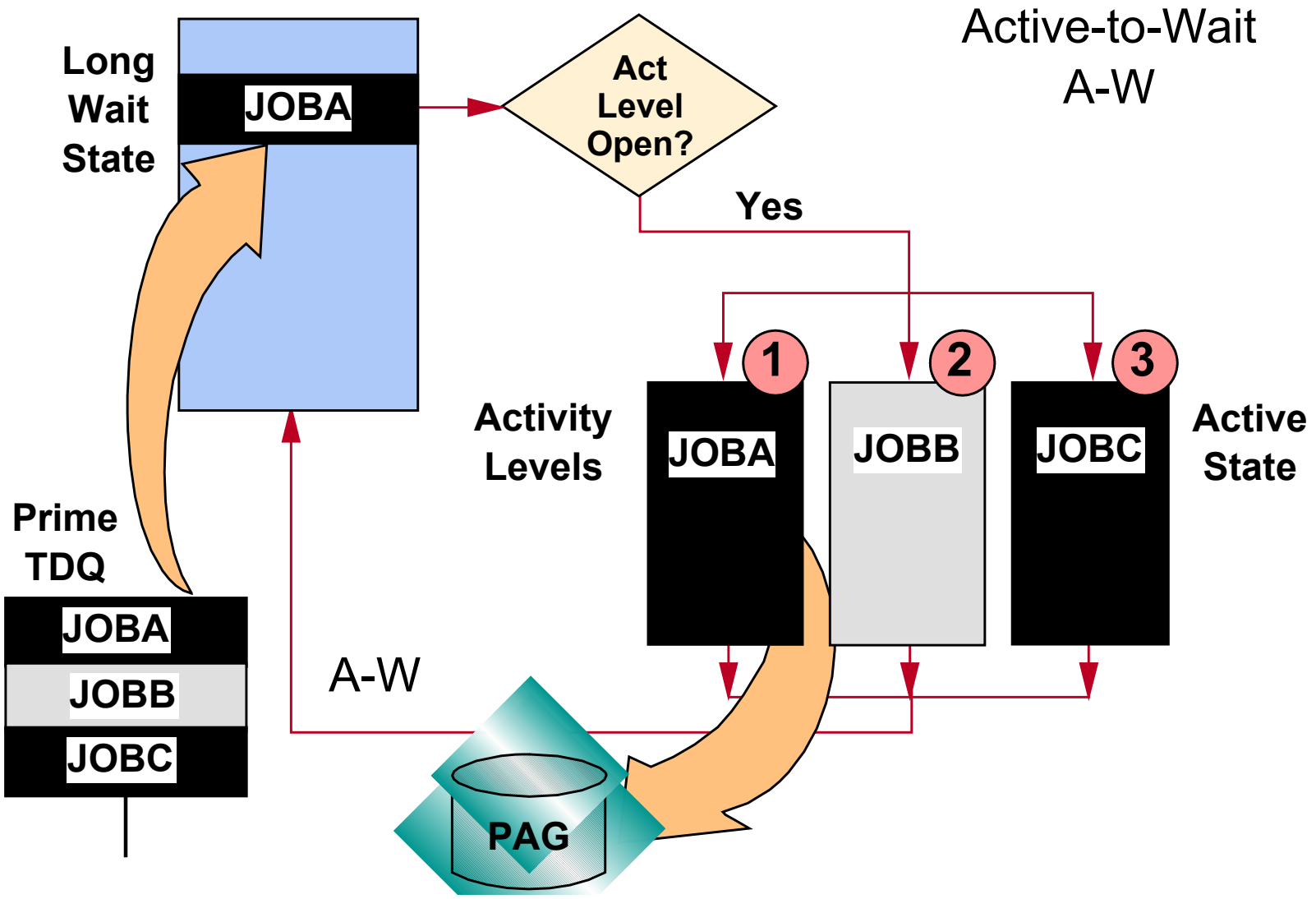
Thread States and Transitions



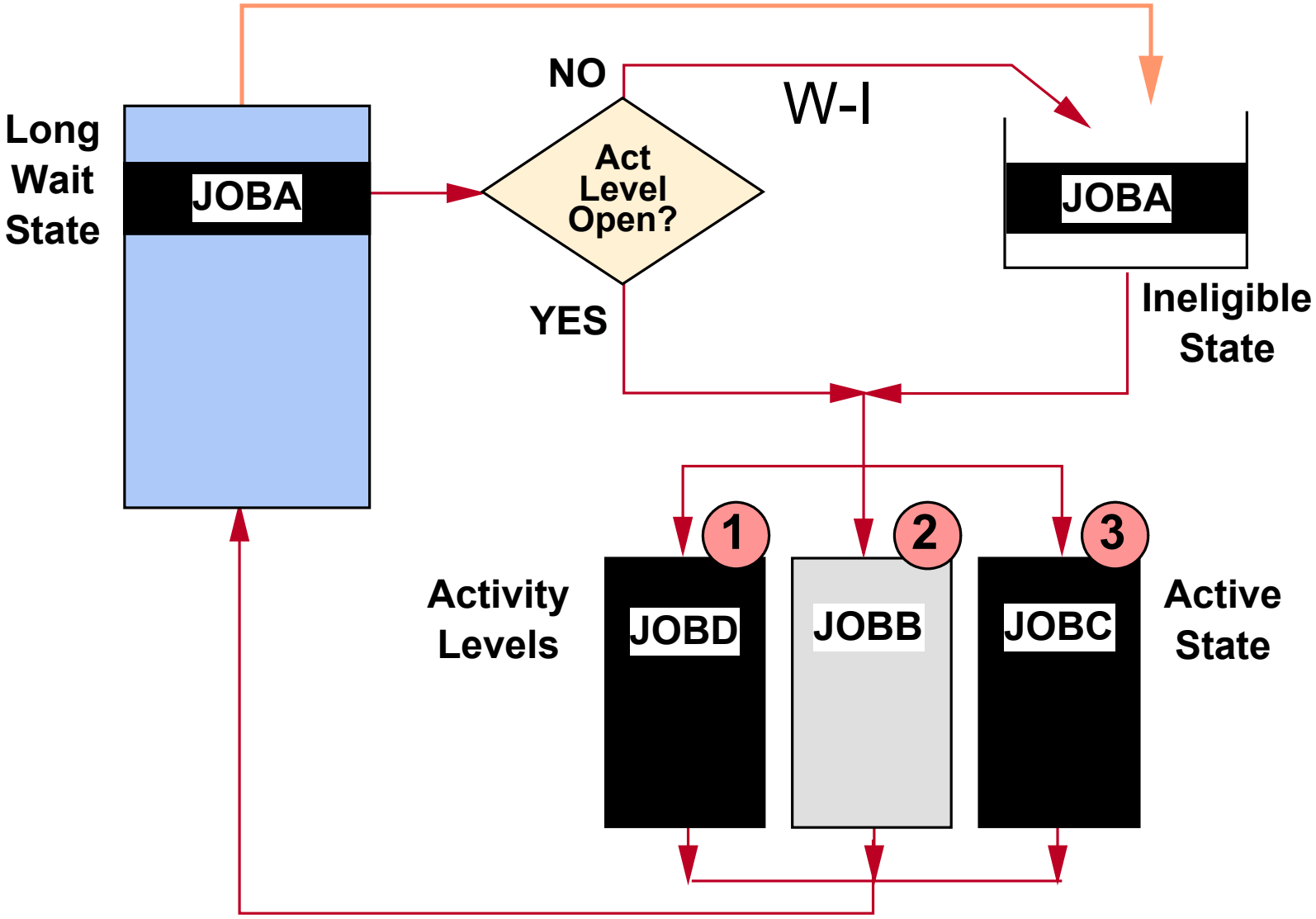
Observing Thread State Transitions

```
                          Work with System Status                         ABC
                                                          06/21/XX        10:11:12
% CPU used . . . . . . :         61.7        Auxiliary storage:
% DB capability . . .  :
Elapsed time . . . . . :      00:06:30             System ASP . . . . . . . . . :    3258 M
Jobs in system . . . . :           494             % system ASP . . . . . . . . :   18.6657
% perm addresses . . . :                           Total  . . . . . . . . . . . :    3258 M
% temp addresses . . . :
   Permanent . . . . .  :         3.063           Current unprotect used . . . :     437 M
   Temporary . . . . .  :         3.962           Maximum unprotect . . . . . .:     437 M

Type changes (if allowed), press Enter.

 System     Pool          Reserved       Max      Active -->Wait --> Active -->
  Pool    Size (M)       Size (M)     Active       Wait      Inel            Inel
    1         3.61           2.26       +++         7.5        .0              .0
    2         3.97              0        10          .4        .0              .0
    3          .23              0         3          .0        .0              .0
    4         8.19              0        25        88.4        .0              .0

                                                                          Bottom
Command
===> ____________________________________________________________________________
4=Prompt      F5=Refresh    F9=Retrieve   F10=Restart
F11=Display pool data  F12=Cancel    F14= Work with subsystems   F24=More Keys
```

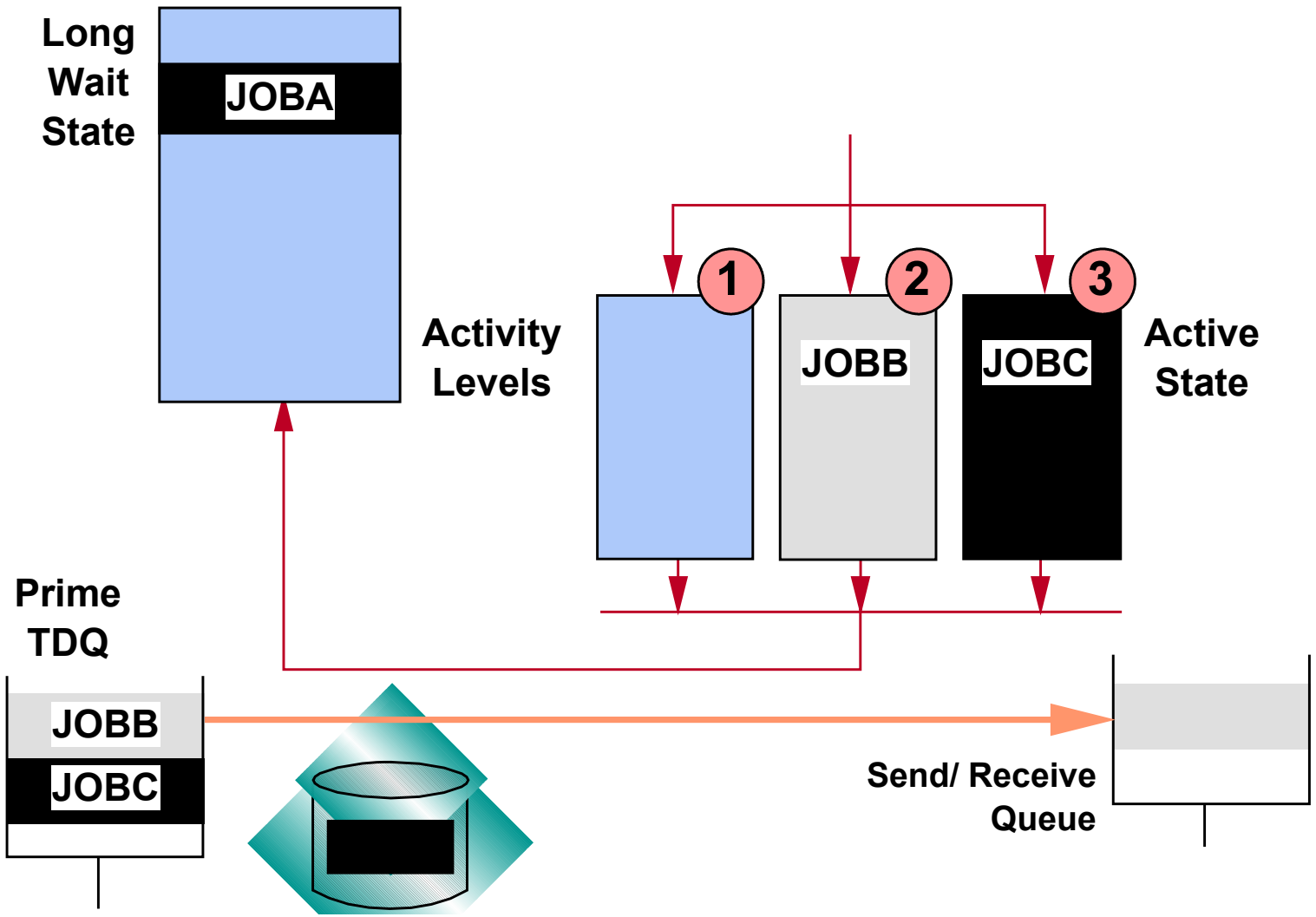
Active-to-Wait



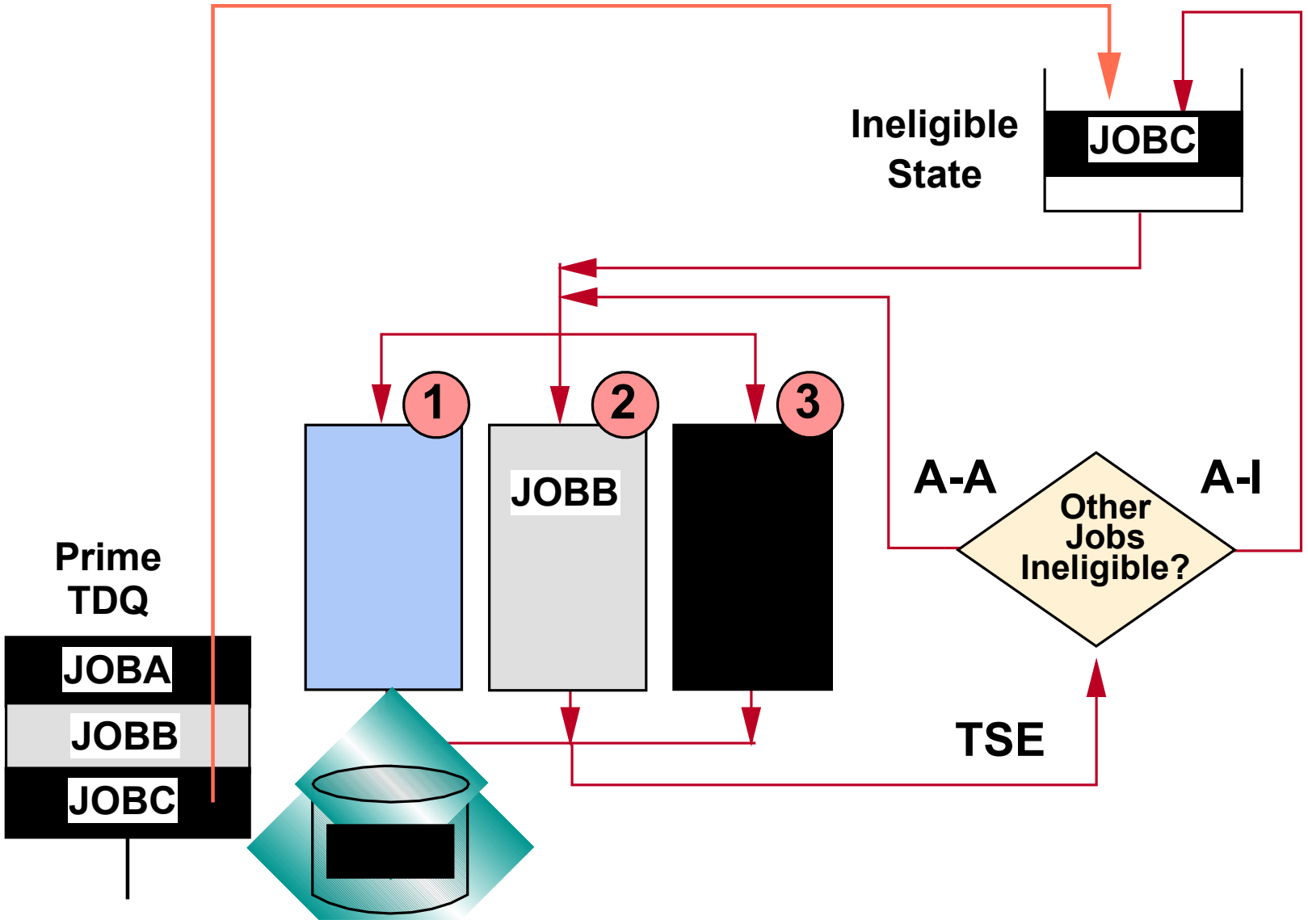
Wait-to-Ineligible



Wait in Activity Level



Active-to-Ineligible



Dynamic Priority Scheduling

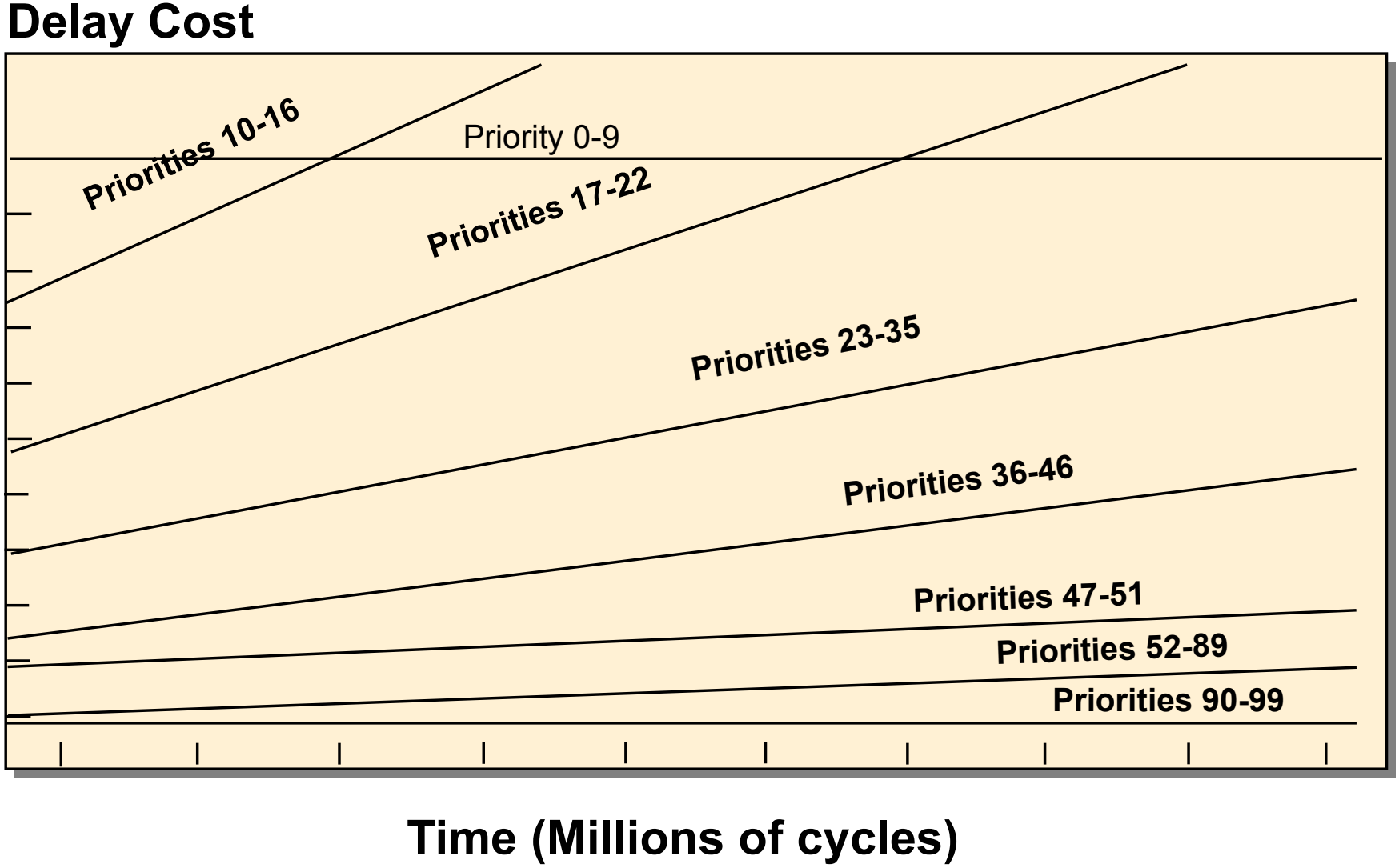
New algorithms on RISC systems

- No job can monopolize the CPU
  - Jobs not using CPU have their DELAY COST increased
- Low-priority jobs have a chance to progress
- iSeries server optimization for batch
- Interactive job priority lowered when batch demand is high

Can be turned on/off

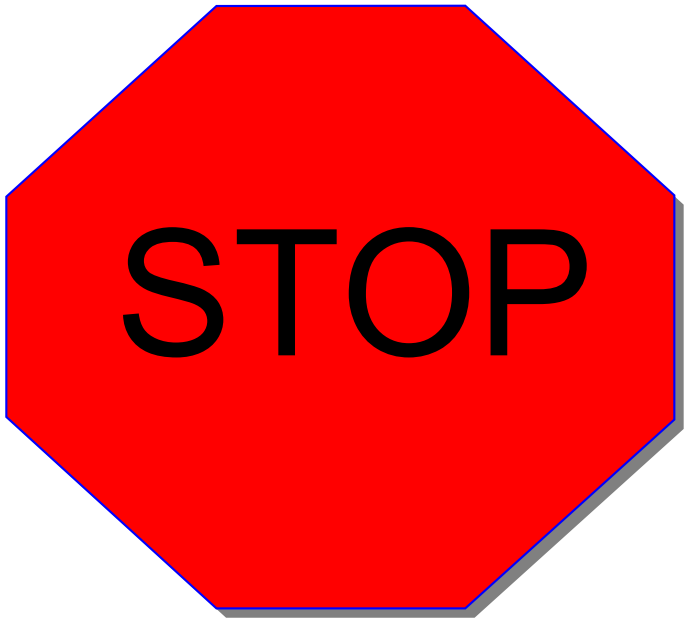
- QDYNPTYSCD and QDYNPTYADJ

Delay Cost Curves



Multi-Processor Considerations (1/2)

Dispatching the first n top TDEs on the TDQ?



Cache Affinity
and
Performance?!

Multi-Processor Considerations (2/2)

Task Dispatching is based on priority, cache affinity and eligibility

Three bits in a TDE

- Eligibility bit
  - one bit for every processor
  - indicates that a task is eligible to run on on corresponding processor
- Active field
  - one bit for every processor
  - indicates on which processor the task is currently running
- Affinity field
  - one bit for every processor
  - Indicates on which processor the task was currently running