Chapter 2

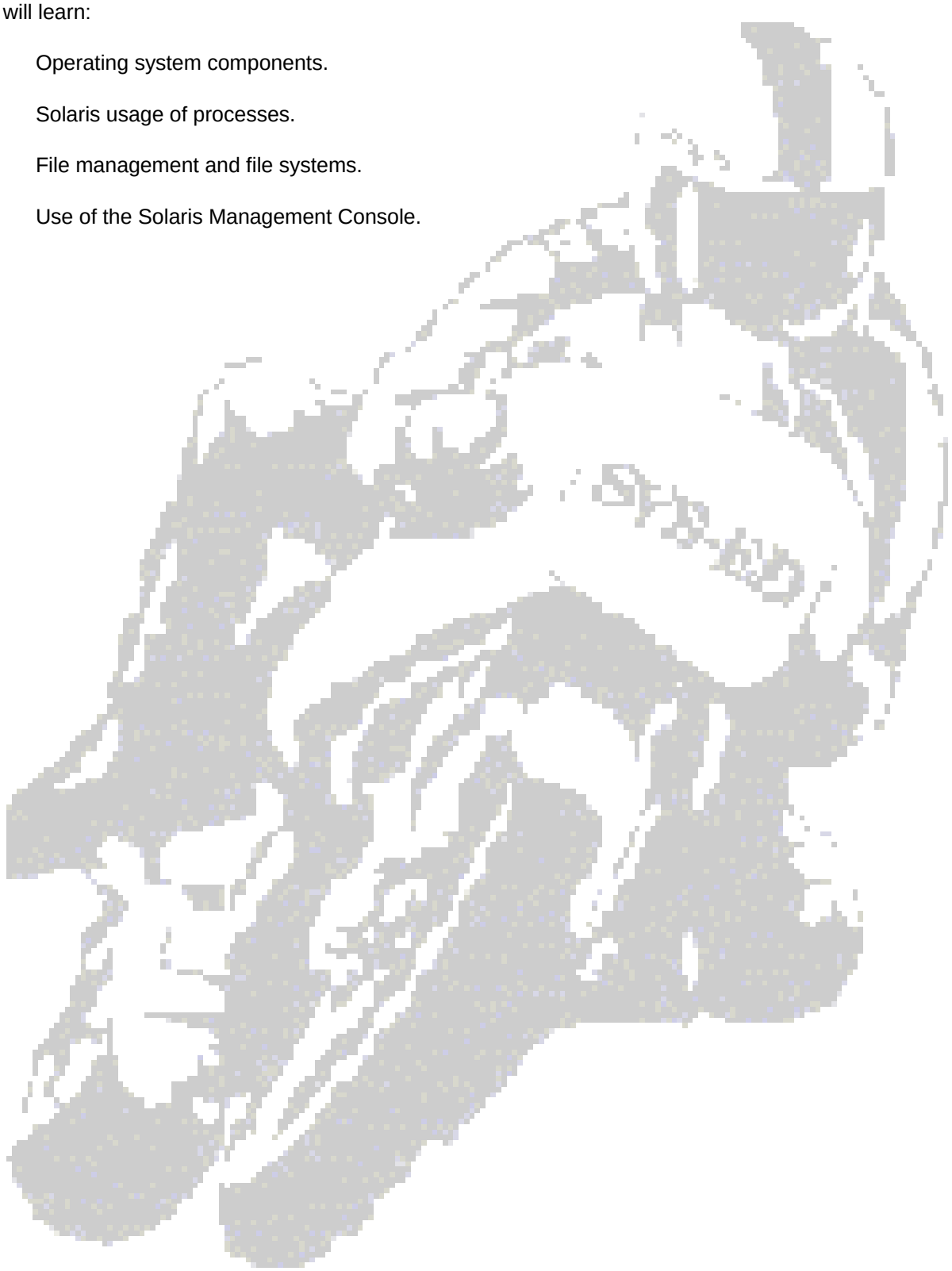
OPERATING SYSTEM CONCEPTS

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- Operating system components.
- Solaris usage of processes.
- File management and file systems.
- Use of the Solaris Management Console.



1 Operating System: Definition

An operating system is the set of programs that controls a computer.

The core of the operating system is the kernel. The kernel is a control program that functions in privileged state that allows all hardware instructions to be executed. It reacts to interrupts from external devices and to service requests and traps from processes. The kernel creates and terminates processes and responds to requests for service.

Operating systems are resource managers. The main resource is computer hardware in the form of processors, storage, input/output devices, communication devices, and data.

Operating system functions include:

- Implementing the user interface.
- Sharing hardware among users.
- Allowing users to share data among themselves.
- Preventing users from interfering with one another.
- Scheduling resources among users.
- Facilitating input/output.
- Recovering from errors.
- Accounting for resource usage.
- Facilitating parallel operations.
- Organizing data for secure and rapid access.
- Handling network communications.

Processes run applications, which are linked together with libraries that perform standard services. The kernel supports the processes by providing a path to the peripheral devices. The kernel responds to service calls from the processes and interrupts from the devices.

Widely used operating systems include:

• UNIX	• Linux

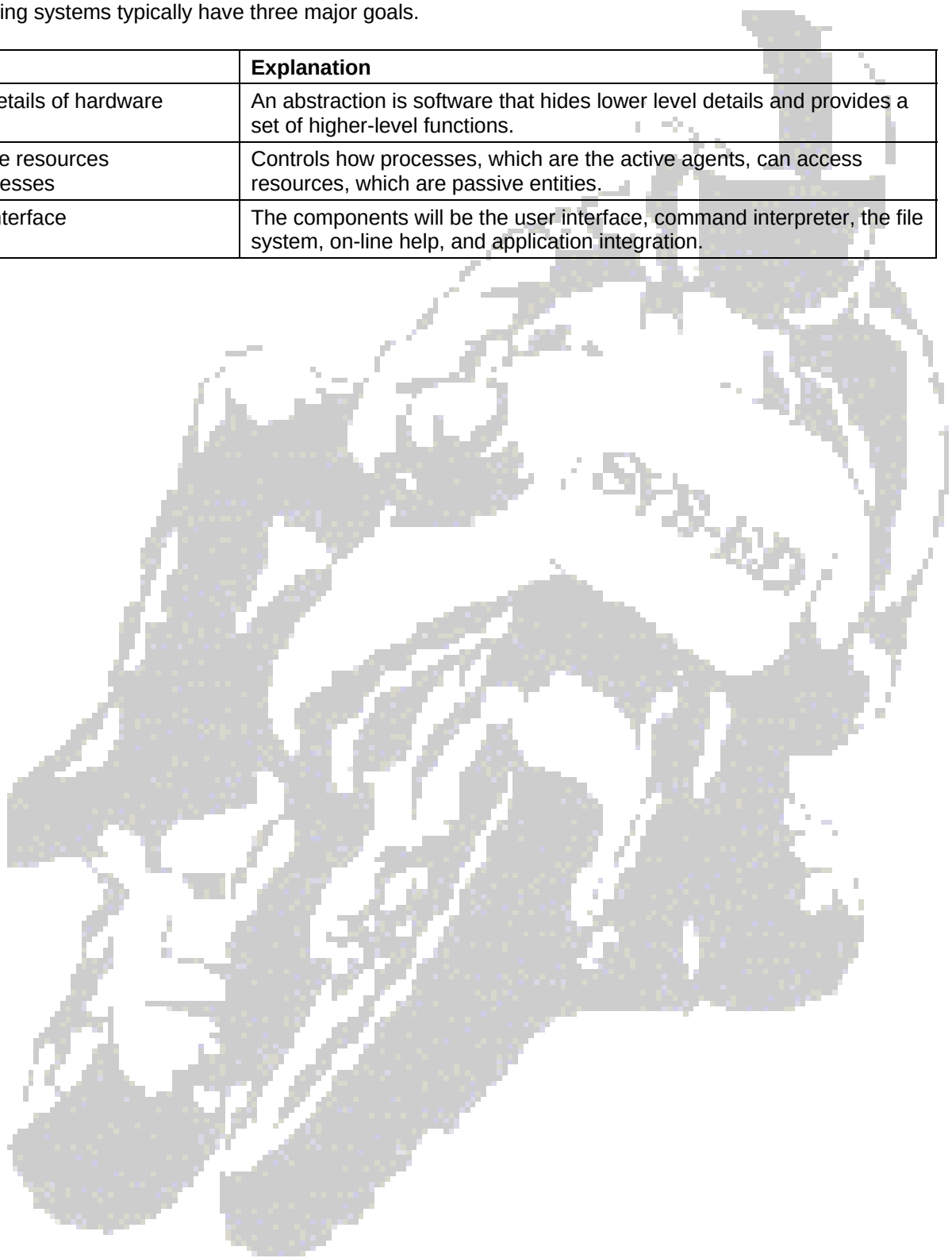
• Microsoft Windows	• MacOS	• z/VM

• VMS	• z/OS

2 Operating Systems: Purpose and Function

Operating systems typically have three major goals.

Goal	Explanation
Hide details of hardware	An abstraction is software that hides lower level details and provides a set of higher-level functions.
Allocate resources to processes	Controls how processes, which are the active agents, can access resources, which are passive entities.
User interface	The components will be the user interface, command interpreter, the file system, on-line help, and application integration.



3 Management Responsibilities

3.1 Process Management

The operating system manages many kinds of activities; these activities are encapsulated in a process which will include the complete execution context such as the code, data, PC, registers, and resources in use, etc.).

A process is not a program; a process is only a single instance of a program in execution. Many processes can be run in the same program.

The major activities of an operating system with respect to process management are:

- Creation and deletion of user and system processes.
- Suspension and resumption of processes.
- A mechanism for process synchronization.
- A mechanism for process communication.
- A mechanism for deadlock handling.

3.2 Main-Memory Management

Primary-Memory or Main-Memory is a large array of words or bytes. Each word or byte has its own address. Main-memory provides storage that can be access directly by the CPU. In order for a program to be executed, it must be in main memory.

The major activities of an operating system with respect to memory-management are:

- Keeping track of which part of memory are currently being used and by whom.
- Deciding which process is loaded into memory when memory space becomes available.
- Allocating and deallocating memory space as needed.

3.3 File Management

A file is a collected of related information defined by its creator. A file systems will be organized into directories; these directories may contain files and other directions.

The major activities of an operating system with respect to file management are:

- Creation and deletion of files.
- Creation and deletion of directions.
- Support of primitives for manipulating files and directions.
- Mapping of files onto secondary storage.
- Back up of files on stable storage media.

The I/O subsystem hides the peculiarities of specific hardware devices from the user. Only the device driver knows the details of the specific device to which it is assigned.

3.4 Secondary - Storage Management

A system will have several levels of storage, including primary storage, secondary storage and cache storage. Instructions and data must be placed in primary storage or cache to be referenced by a running program. Main memory will be too small to accommodate all data and programs, and its data will be lost when power is lost. Accordingly, the computer system must provide secondary storage to back up main memory.

Secondary storage consists of tapes, disks, and other media designed to hold information that will eventually be accessed in primary storage. It will be divided into bytes or words consisting of a fixed number of bytes. Each location in storage has an address; the set of all addresses available to a program is called an address space.

The three major activities of an operating system with respect to secondary storage management are:

- Managing the free space available on the secondary-storage device.
- Allocation of storage space when new files have to be written.
- Scheduling the requests for memory access.

3.5 Networking

A distributed system is a collection of processors that do not share memory, peripheral devices, or a clock. The processors communicate with one another through communication lines. The communication-network design must account for routing and connection strategies, and the problems of contention and security.

3.6 Program Execution

The purpose of a computer system is to allow the user to conveniently execute programs.

Running a program involves the allocation and deallocation of memory. It also involves in the case of multiprocessing CPU scheduling. User-level programs cannot help the user to run programs independently without the help from operating systems.

3.7 I/O Operations

Each program requires an input and produces output. The operating systems hides the details of the underlying hardware for the I/O. All the user sees is that the I/O has been performed without any details. By providing I/O, the operating system makes it convenient for the users to run programs.

Users cannot control I/O and this service cannot be provided by user-level programs.

3.8 File System Manipulation

The output of a program may need to be written into new files or input taken from some files. The user will invoke a command for reading or writing to a file and the task will be accomplished. The user does not have to worry about secondary storage management. The operating systems provides this service.

This service will involve secondary storage management. The speed of I/O will in most situations be dependent on secondary storage management . Although it is not difficult for user-level programs to provide these services, it will be best if this service is left with the operating system.

3.9 Communications

There are instances where processes need to communicate with each other to exchange information. It can be between processes running on the same or different computers.

By providing this service, the operating system relieves the user from the responsibility of passing messages between processes. In the situation, where the messages need to be passed to processes on the other computers through a network it can be done by the user programs. The user program can be tailored to the hardware through which the message tranmsits and provides the service interface to the operating system.

3.10 Error Detection

An error is a part of the system which may cause malfunctioning of the complete system. In order to avoid such a situation, the operating system monitors the system for detecting the errors. This relieves the user from the concern as to whether errors will propagate to other areas of the system and cause additional problems.

Error detection service, cannot allow to be handled by user programs because it involves monitoring and in some cases altering area of memory or deallocation of memory for a faulty process. A user program if given these privileges can interfere with the normal operation of the operating systems.

4 Process: Definition

A process is fundamental to understanding how an operating system functions. Process is commonly used interchangeably with 'task' or 'job'.

Common accepted definitions include:

- A program in execution.
- An asynchronous activity.
- The 'dispatchable' unit.

4.1 Process: Different from a Program

Process is not identical to a program; it is more than program code. A process is an 'active' entity as compared to a program which is a 'passive' entity.

A process includes:

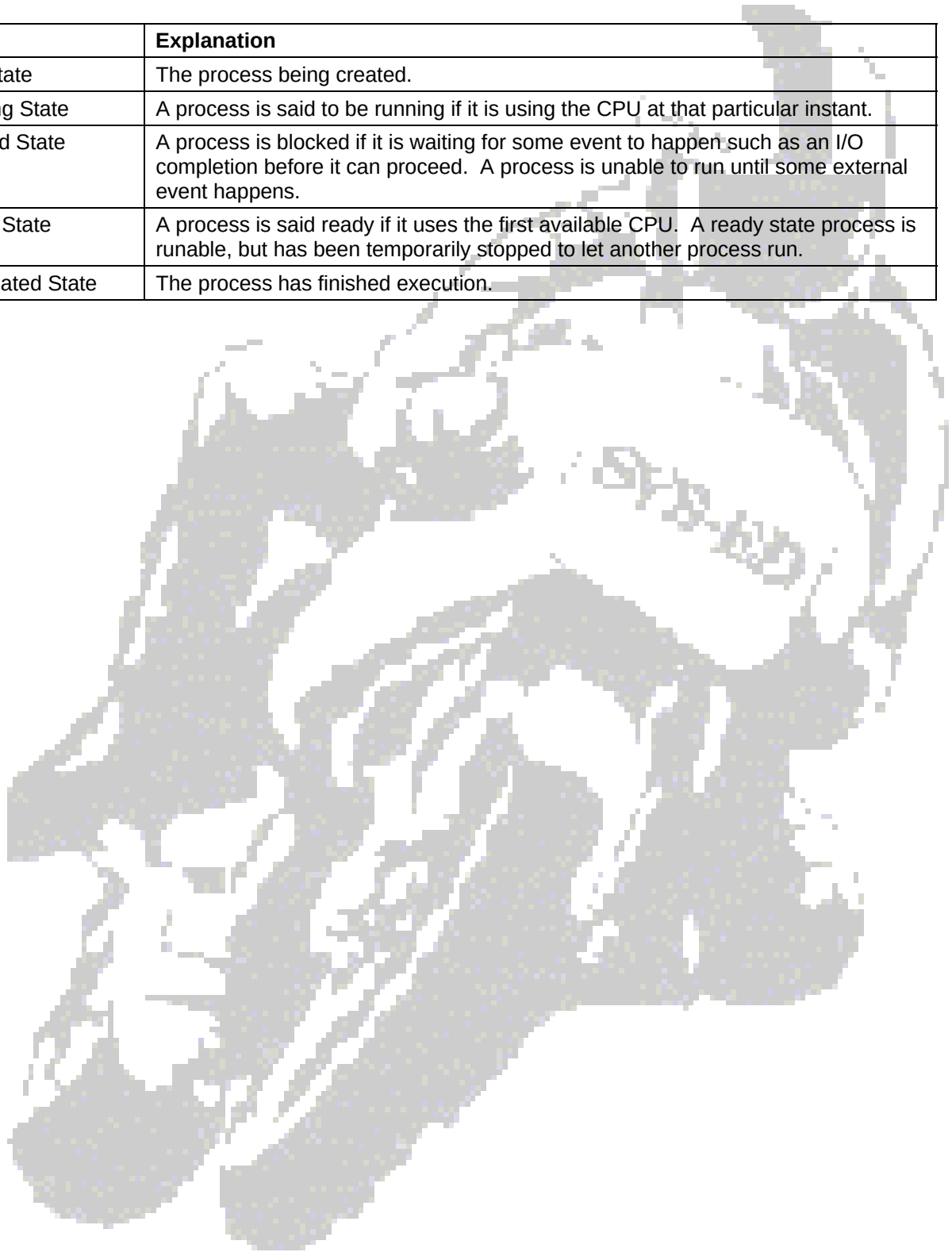
- Current value of Program Counter.
- Contents of the processors registers
- Value of the variables
- The SP: process stack will contain temporary data such as subroutine parameter, return address, and temporary variables.
- A data section that contains global variables.
- A process is the unit of work in a system.

In Process model, all software on the computer is organized into a number of sequential processes. A process includes PC, registers, and variables. Each process has its own virtual CPU. The CPU switches back and forth among processes; this is multiprogramming.

4.2 Process State

A process goes through a series of discrete process states.

State	Explanation
New State	The process being created.
Running State	A process is said to be running if it is using the CPU at that particular instant.
Blocked State	A process is blocked if it is waiting for some event to happen such as an I/O completion before it can proceed. A process is unable to run until some external event happens.
Ready State	A process is said ready if it uses the first available CPU. A ready state process is runable, but has been temporarily stopped to let another process run.
Terminated State	The process has finished execution.



5 CPU/Process Scheduling

The scheduling algorithms can be divided into two categories with respect to how they deal with clock interrupts.

5.1 Nonpreemptive Scheduling

A scheduling discipline is nonpreemptive if, once a process has been given the CPU, the CPU cannot be taken away from that process.

With nonpreemptive scheduling:

- Short jobs are made to wait by longer jobs, but the overall treatment of all processes is fair.
- Response times are more predictable because incoming high priority jobs can not displace waiting jobs.
- A schedular executes jobs in the following two situations:
  - When a process switches from running state to the waiting state.
  - When a process terminates.

5.2 Preemptive Scheduling

A scheduling discipline is preemptive if a process has been given the CPU, which can then be taken away. This is in contrast to the run to completion method.

6 Solaris Management Console

The Solaris Management Console includes a default toolbox for:

- managing users
- projects
- cron jobs
- mounting and sharing file systems
- managing disks and serial ports.

Tools can be added to the existing toolbox or new toolboxes can be created.

The Solaris Management Console has three primary components:

• Console Client	• Console Server	• Console Toolbox Editor

7 Solaris Management Console Tools

Category	Tool	Description
System Status	System Information	Monitors and manages system information such as date, time, and time zone.
	Log Viewer	Monitors and manages the Solaris Management Console tools log and system logs.
	Processes	Monitors and manages system processes.
	Performance	Monitors system performance.
System Configuration	Users	Manages users, rights, roles, groups, and mailing lists.
	Projects	Creates and manages entries in the /etc/project database.
	Computers and Networks	Creates and monitors computer and network information.
	Patches	Manages patches.
Services	Scheduled Jobs	Creates and manages scheduled cron jobs.
Storage	Mounts and Shares	Mounts and shares file systems.
	Disks	Creates and manages disk partitions.
	Enhanced Storage	Creates and manages volumes, hot spare pools, state database replicas, and disk sets.
Devices and Hardware	Serial Ports	Sets up terminals and modems.

8 UNIX System Administration

Systems administration is the installation and maintenance of the hardware and software on the UNIX computer system. This will typically entail hardware configuration, software installation, reconfiguration of the kernel, and networking tasks.

In order to perform these tasks, it will be necessary to assume superuser, or root, privileges to perform the tasks which are not available to the average user of the system.

8.1 Common Tasks

- Accounts are added by assigning login id's, groups, user id numbers, group id numbers, login directories, and set-up the users' login environments. Other related tasks include balancing the needs of various users with quotas on disk space or limits on simultaneous processes.
- Monitor disk status, system processes, user process activity, system security, and system log files in order to ensure that resources are available and that only valid users have access.
- Manage disk space usage, tape and CD ROM devices and network services.
- Backup and restore procedures for insuring data integrity against disk crashes, users accidentally deleting files, for the removal of seldom used programs to free up disk space, etc.
- Quotas restrict users to a finite disk space and can be set individually.
- Keeping the system running and providing maintenance.