

I *In the Beginning*



1 What Is Reliability?

Historically, the UNIX operating system has been for the computer cognoscenti and has seen a lot of use in research, operating system and application development, engineering, etc. In recent years, UNIX systems have become an integral part of many organizations' business operations—many UNIX systems are now “mission critical” production systems. As a result, the need for reliable and repairable UNIX systems has never been greater.

Much of what is discussed in this booklet is UNIX-centric, but the principles and goals apply to systems running any operating system. So while many of the examples assume a UNIX environment, remember that the lessons apply to other systems, such as Windows NT, PC desktops, and anything else you might have in your environment.

1.1 Definition

If we're going to have any chance of ending up with reliable systems, we must first define what we mean by reliability. The most obvious definition is that:

A “reliable” system is one that's always working properly when I want to use it.

Instead of that simpleminded definition, let's try this one:

A “reliable” system is one that is configured and maintained to provide the appropriate levels of Redundancy, Repeatability, and Repairability, in order to provide an appropriate level of support for the services that the system is intended to provide.

(The Key Principles of Redundancy, Repeatability, and Repairability are defined below in section 1.2, and the difference between services, servers, and systems is covered in section 2.1.)

Now, that's not a very exact definition—it contains two “appropriate”s and one “intended”; it doesn't seem to be very rigorous. But the point is that systems administration and configuration, like many other things, is a constant series of trade-offs and compromises, and that each decision has to be made based on the needs of a particular organization for a particular service. The most obvious trade-off you can make is between reliability and money—you can typically get “reasonable” reliability at a “reasonable” cost (uh oh, more weasel words), but to get high reliability you often have to pay a high price. But there are lots of other trade-offs made, including those between security and accessibility, performance and redundancy, and between providing 100% uptime during the day and not having to work every evening and every weekend.

Two more introductory points. This booklet is primarily about “higher reliability” rather than “high reliability.” The emphasis is on the incremental steps that can be

2 / What is Reliability?

taken to improve reliability in your systems, rather than on building 100% uptime, absolutely no-failure, no-compromise systems. And second, if you need a quick one-word summary to provide to your boss, use this one: RAS—reliability, availability, serviceability.¹

Do you need higher reliability? As always, it's a matter of weighing the costs and benefits for your particular situation. For example, your home workstation may not need higher reliability, unless you sell Web space on it or provide other services, in which case you should consider the effects of system downtime on your customers and your business. In most organizations it will be the server systems that may benefit from higher reliability—client machines (e.g., desktop workstations) are less likely to justify the increased cost of higher reliability (not always though—consider the case of workstations used by financial market traders).

You will need to examine the services required or provided by your organization, and the nature of the organization itself, in order to identify where you may need to consider increased reliability. For example, what happens if:

- You have a disk failure on your file server?
- The network interface on your DNS server stops working?
- You have a power failure in your building?

Sometimes a failure in the simplest services can have significant effects on your operations—you will need to consider how your servers and services interact.² The nature of your organization will often dictate how reliable your systems must be. For example, if you're a network or computing service provider, you may wish to have very reliable systems, in order to provide excellent service and retain your customers in a very competitive market.

1. RAS is often thought of in terms of service metrics for measuring achievement. More on metrics in section 1.4.

2. The distinction between servers and services is often a useful one to make. Servers are typically hardware (computers, network equipment, etc.), and services are what the hardware is used to provide, e.g., mail, DNS, web, file, print and other services. See sections 2.1 and 3.2 for more information.