CHAPTER 5

CPU Scheduling

CPU scheduling is the basis of multiprogrammed operating systems. By switching the CPU among processes, the operating system can make the computer more productive. In this chapter, we introduce basic CPU-scheduling concepts and present several CPU-scheduling algorithms, including real-time systems. We also consider the problem of selecting an algorithm for a particular system.

In Chapter 4, we introduced threads to the process model. On modern operating systems it is kernel-level threads—not processes—that are in fact being scheduled by the operating system. However, the terms “process scheduling” and “thread scheduling” are often used interchangeably. In this chapter, we use process scheduling when discussing general scheduling concepts and thread scheduling to refer to thread-specific ideas.

Similarly, in Chapter 1 we describe how a core is the basic computational unit of a CPU, and that a process executes on a CPU’s core. However, in many instances in this chapter, when we use the general terminology of scheduling a process to “run on a CPU”, we are implying that the process is running on a CPU’s core.

Bibliographical Notes

Feedback queues were originally implemented on the CTSS system, described in [Corbato et al. (1962)]. This feedback queue scheduling system was analyzed by [Schrage (1967)]. The preemptive priority scheduling algorithm of 5.24 was suggested by [Kleinrock (1975)]. The scheduling algorithms for hard real-time systems, such as rate monotonic scheduling and earliest-deadline-first scheduling, are presented in [Liu and Layland (1973)].

[Anderson et al. (1989)], [Lewis and Berg (1998)], and [Philbin et al. (1996)] discuss thread scheduling. Multicore scheduling is examined in [McNairy and Bhatia (2005)] and [Kongetira et al. (2005)].

[Fisher (1981)], [Hall et al. (1996)], and [Lowney et al. (1993)] describe scheduling techniques that take into account information regarding process execution times from previous runs.

Scheduling policies used in the UNIX V operating system are described by [Bach (1987)]; those for UNIX FreeBSD 5.2 are presented by [McKusick et al. (2015)]; and those for the Mach operating system are discussed by [Black (1990)]. [Love (2010)] and [Mauerer (2008)] cover scheduling in Linux. [Faggioli et al. (2009)] discuss adding an EDF scheduler to the Linux kernel. [Lozi et al. (2016)] evaluates the current Linux scheduler with respect to multicore and NUMA systems. Details of the ULE scheduler can be found in [Roberson (2003)]. Solaris scheduling is described by [Mauro and McDougall (2007)]. [Russinovich et al. (2017)] discusses scheduling in Windows internals. [Butenhof (1997)] and [Lewis and Berg (1998)] describe scheduling in Pthreads systems. [Siddha et al. (2007)] discuss scheduling challenges on multicore systems.

Bibliography


Bibliographical Notes 169


Chapter 5  CPU Scheduling