

(三)简答题

1. 从结构上看, UNIX 由哪两部分组成?分别包含哪些内容?
2. UNIX 系统中, 用户程序如何请求系统服务?操作系统又如何处理?
3. UNIX 中什么是标准输入 / 输出?什么是输入输出重定向?
4. 什么是管道技术?它与输入输出重定向有何不同?
5. 简述执行 shell 文件的两种方法。
6. UNIX 进程与通常操作系统的进程有何不同?
7. 试述 UNIX 进程的组成。
8. 用 fork 创建新进程, 它要做哪些工作?
9. UNIX 系统中的优先权和优先数有什么关系?如何确定进程的优先权和优先数?
10. UNIX 中采用对换技术有何作用?
11. 在页面调度时, UNIX 采取了哪些优化措施?
12. UNIX 文件系统由哪两个部分组成?
13. UNIX 文件的物理结构采用什么方式?它有什么特点?
14. UNIX 采用什么技术实现块设备的读写操作?
15. UNIX 对缓冲区队列管理时, 何时能释放缓冲区?被释放的缓冲区链入空闲队列, 同时仍保留在设备缓冲区队列, 起何作用?
16. 在缓冲区控制块中对缓冲区设置 DELWR 标志表示什么?怎样才能分配这种缓冲区做其他用途?
17. 字符设备管理时采用什么缓冲技术?

(四)应用题

1. 假设进入 UNIX 系统后, 当前正处在根目录下, 若 / home / user1 中已有若干文件。该目录允许所有人具有读、写、执行的权利, 试用 shell 命令实现以下功能: (1)使当前目录变成 / home / user1; (2)显示当前目录; (3)在该目录下建一目录 newdir; (4)显示当前目录中的文件名; (5)将 abC. C 在 / home / user1 / newdir 下做一个备份; (6)显示 / home / user1 / newdir / abC. C 的内容; (7)删除 / home / user1 / newdir / abC. C; (8)删除 / home / user1 / newdir 目录。
2. 在当前目录 / usr / student 下已有若干文件, 你对此目录及其中的所有文件拥有读、写、执行的许可权, 若现在已在 C shell 下工作, 请完成: (1)把文件 data 作为程序 ex1 的输入, 运行 ex1; (2)把文件 a 和文件 b 合并成文件 c (3)把文件 d 加在文件 c 的尾部; (4)对 1etter 文件做拼写检查, 并转后台执行; (5)打印一份按反序排列的文件清单; (6)把反序排列的文件清单保存到文件 f_ls 中。
3. 用 C shell 编写一个 shell 文件 script, 并执行之。(1) script 的要求是: 对两个 C 语言程序分别进行编译, 并改名为 ex1 和 ex2; 把一文件作为 ex1 的输入, 运行 ex1, 并把其输出作为 ex2 的输入, 运行 ex2 且把结果存入另一个文件。要求此 script 为一个通用的程序, 允许使用时用不同的文件代入。(2)执行此 script 条件是两个 c 语言程序为 ex1.c 和 ex2.c, ex1 的输入为 data1 文件, ex2 的输出存入 data2 文件。

(四)应用题

1. (1)cd / home / user1 (2)pwd (3)mkdir newdir (4)ls (5)cp abc.c. / newdir (6)cat. / newdir / abc.c (7)rm. / newdir / abc.c (8)rmdir newdir
2. (1)ex1<data (2)cat a b>c (3)cat d>>c (4)spell letter & (5)ls | sort -r | lpr (6)ls | sort -r>-f-ls
3. (1)script 文件如下:
cc \$1
mv a.out ex1
cc \$2
mv a.out ex2
ex1<\$3 | ex2>\$ 4
(2)csch sCRIPT ex1,c ex2.c data1 data2

(三)简答题

1. 从结构上看, UNIX 可以分成内核层和外壳层两部分。内核层是 unix 的核心, 它实现存储管理、文件管理、设备管理、进程管理等功能, 并为外壳层提供系统调用。外壳层为用户提供各种操作命令和程序设计环境。外壳层由 shell 解释程序、支持程序设计的各种语言的编译程序和解释程序、实用程序和系统库等组成。
2. UNIX 规定用户程序用“trap 指令”请求系统服务。当处理器执行到 trap 指令时形成一个中断, 然后由内核的 trap 处理子程序根据 trap 指令中的系统调用分类号转向相应的处理程序。当系统调用处理程序完成处理后, 先返回到 trap 处理子程序, 由 trap 处理子程序进行后续处理后, 再返回用户程序。
3. 从键盘上读入信息称为标准输入, 把信息显示到屏幕上称为标准输出。输入输出重定向是指把标准输入改成从文件中读信息, 把标准输出改成将信息写到文件中去。
4. UNIX 系统采用管道技术实现进程间的信息传送, 管道实际上是连接在进程间的可共享文件, 称为 pipe 文件, 进程可对 pipe 文件进行读、写操作。管道把一个进程的标准输出与另一个进程的标准输入连接起来。管道技术不必建立中间文件作媒介, 而输入输出重定向技术实现进程间传递信息时, 必须要建立一个中间文件, 比如 p1>file, 把进程 p1 的输出重定向到文件 file 中, 再执行 p2<file, 使 file 作为进程 P 2 的输入。当然最后还得把 file 删除。
5. 假设已写成的 shell 文件为 proc, 要执行时可用以下两种方法。
(1)使用 csh 命令: csh proC (2)把 proc 改成可执行文件后直接执行: chmod +x proc
proc

注: 这里 proc 无参数, 若需参数执行时可加在其后。

6. 通常操作系统把进程分成两大类, 一类是系统进程, 它执行操作系统程序, 提供系统服务; 另一类是用户进程, 它执行用户程序, 完成用户功能。而 UNIX 进程既可执行系统程序, 也可执行用户程序, 按照需要进行转换。为此 UNIX 进程有两种执行环境, 即核心态和用户态。操作系统程序在核心态执行, 用户程序在用户态执行。
7. UNIX 进程有三部分组成: 进程控制块、正文段和数据段。进程控制块的一部分为基本控制块, 它常驻内存, 记录了进行进程调度时必须使用的主要信息; 另一部分为扩充控制块, 它不常驻内存, 随用户程序和数据换进或换出主存。正文段是可供多个进程共享的程序, 它由不可被修改的程序和常数组成。数据段是指进程执行时用到的数据, 若进程执行的子程序是非共享的, 也包含在数据段内。
8. 由 fork 创建新进程的主要工作有: (1)在进程表 proC[]中为子进程找一个空闲的表项, 用来存放子进程的 proc 结构。(2)为子进程分配一个唯一的标识号。(3)把父进程中的字段复制到子进程的 proc 中, 并把 p_pid 置为分配到的进程标识号, 把 p_ppid 置为父进程的标识号, 把 p_star 置为创建状态。(4)按父进程中 P_size 所示的长度为子进程申请分配内存。若有足够的内存, 则把父进程的 user 结构、栈和用户数据区全部复制到子进程的空间中, 若无足够的内存, 则在磁盘对换区中分配存储空间, 然后复制到对换区中, 置子进程状态为就绪状态。
9. UNIX 中每个进程都有一个优先数, 就绪进程能占用处理器的优先权取决于进程的优先数, 优先数越小则优先权越高。UNIX 以动态方式确定优先权, 如核心态的进程优先权高于进入用户态的进程; 降低用完一个时间片的进程的优先权; 对进入睡眠的进程, 其等待事件越急优先数越高; 降低使用处理器时间较长的进程的优先权。UNIX 中确定进程优先数的方法有两种: 设置方法和计算方法。前者对要进入睡眠状态的进程设置优先数, 若等待的事件急迫, 则设置较小的优先数; 后者用户进程正在或即将转入用户状态运行时确定优先数。
- 10 对换区可作为内存的逻辑扩充, 用对换技术解决进程之间的内存竞争。

11. 为提高系统效率, UNIX 采取了以下优化措施: (1)不能淘汰正在与设备交换信息的页面和正在被装入的页面; (2)在页表中设置有效位, 使无效页面所对应的内存块尚未移作他用时, 进程仍对访问 (3)采用二次机会页面替换算法; (4)空闲页面被重新分配时, 若它未被修改过, 就可不必写回磁盘; (5)用页面守护进程监视空闲页面个数, 减少缺页中断次数; (6)当系统负载过重时, 采用交换技术, 腾出内存空间。
12. UNIX 文件系统有基本文件系统和可装卸的子文件系统两部分。基本文件系统是 UNIX 文件系统的基础, 系统启动后它不能脱卸, 而子文件系统可以随时更换。这种结构使文件系统易于扩充和更改。
13. UNIX 文件的物理结构采用索引结构方式。每个文件都有一张索引表记录文件占用的磁盘空间块号, 每个索引表中使用 13 个索引项, 前十个索引项直接指向文件数据在磁盘中的块号, 后三个索引项分别是一次间接、二次间接和三次间接的索引项。采用这种间接索引既缩短了索引节点的长度, 又大大增长了文件的许可长度。
14. UNIX 采用缓冲技术实现块设备的读写操作。驱动程序为写操作分配一个缓冲区, 先从用户空间把信息复制到缓冲区, 再从缓冲区输出到设备; 对读操作, 先从设备接收信息到缓冲区, 再将信息复制到指定的用户空间。采用缓冲技术减少对块设备的访问次数可提高系统的吞吐量。
- 15 当缓冲区的信息传送到用户空间后, 或用户信息写到缓冲区且对缓冲区置了 DELWR 标志后, 这样的缓冲区就可以释放。把它链入空闲缓冲区队列的目的是为了使缓冲区能被充分利用, 需要时作为空闲缓冲区分配; 把它仍保留在设备缓冲区队列的目的是需要时可以继续利用缓冲区的信息。
16. 对块设备缓冲区设置 DELWR 标志表示缓冲区的内容尚未写到磁盘上。如果要分配这个缓冲区作其他用途时, 必须先把该缓冲区数据块中的内容写到磁盘上。
17. 字符设备管理时采用缓冲区较小、易管理的字符缓冲区技术。缓冲区的容量大约为 64 个字符。UNIX 对字符缓冲区采用单链结构, 系统初始启动时, 所有的字符缓冲区都链在空闲缓冲区队列中, 每次分配和释放缓冲区都在队首进行。