山东大学 软件 学院

Linux应用 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202200201095 | 姓名： 杨伟康 | | 班级： 22网安 |
| 实验题目： 实验二：文件系统 | | | |
| 实验学时：4 | | 实验日期： 10.30 | |
| 实验目的：  1．掌握文件、文件的类型和文件系统的结构  2．熟练掌握对文件和目录的操作。  包括：  显示文件列表路径命令：（ls、pwd）  创建文件和目录：(cd、mkdir、touch)  移动、复制和删除文件和目录：（copy、mv、rm、rmdir）  符号连接命令：（ln）  查看文件内容：（用cat、more、less命令）  3、掌握对文件和目录权限的操作  相关命令：umask、chmod  4、熟悉其他命令的使用：  其它命令包括：（wc、du、df） | | | |
| 硬件环境：  PC机 | | | |
| 软件环境：  VMware Workstation 17 Player+ Ubuntu 22 | | | |
| 实验过程报告：  一. 对目录的操作   1. 检查你当前所在的目录，该目录是 \_/home/yang/桌面\_\_\_\_\_\_\_\_\_.      1. 改变当前目录到根目录(/).      1. 验证是否在根目录，分别用简单列表和长列表的方式列出目录中的文件.   简单列表  长列表    4.列出当前目录中的所有文件，列出当前目录及其子目录中的所有文件.  注意: 该命令会有大量的输出. 要停止的话,可以用 <Ctrl> +c来中止.  这里只截取其中的一部分    5. 返回到你的主目录，列出该目录的内容包括隐藏文件.    6. 在你的主目录里创建一个名为mydir的目录. 然后, 发出命令分别显示你的主目录和~/mydir目录的长列表. (不显示目录下的内容.) 每一个目录的大小是多少? \_4096\_\_\_\_\_\_\_\_\_    7. 分别用绝对路径和相对路径两种方式转到mydir目录. 创建两个名为myfile1和myfile2的文件.    8. 用长列表形式显示mydir目录的内容. 长列表列出的各栏分别显示了什么信息?文件myfile1和myfile2的大小是多少?大小是0    文件类型和权限：首列展示了文件的类型（例如，-代表普通文件，d代表目录）以及文件的权限（例如，rwxr-xr-x）。  硬链接数：第二列显示了指向该文件的硬链接数量。  文件所有者：第三列显示了文件的拥有者用户名。  文件所属组：第四列显示了文件所属的组名。  文件大小：第五列以字节为单位显示了文件的大小。对于目录，这个数字通常表示目录本身在文件系统中的大小，而非其内容的总大小。  时间戳：接下来的三列（可能合并为两列，具体取决于终端宽度）分别展示了文件的最后访问时间、最后修改时间和最后状态改变时间。然而，这些时间戳的具体显示可能受到ls命令选项和文件系统特性的影响。  文件名：最后一列显示了文件的名称。  9. 返回到你的主目录然后使用 ls -R 命令来显示你的目录树. 解释显示出来的信息。    显示出来主目录下的所有文件，以及各个文件夹里面的内容  10. 试着用rmdir删除mydir目录. 能行吗?为什么?    12. 进入mydir目录删除该目录中的2个文件. 然后返回到你的主目录并且删除掉mydir目录.    13. 用一条rm命令可否实现12题中的要求？如何实现？解释rm命令的作用。  **使用单条 rm 命令确实可以实现第12题中的所有要求，但需要注意的是，这种操作具有潜在的风险，因为它会一次性删除指定的文件和目录，而不提供逐个确认的机会。在实现这一目标时，我们必须确保所指定的路径和文件名都是准确无误的。**  二. 对文件的操作  14. 分别用cat,more和less命令查看 /etc/passwd 文件的内容。/etc/passwd 文件包含了能使用该系统的所有授权的用户的一个列表.        这三个命令显示的方式有什么不同？用more命令查看时，验证以下按键的功能：  f键或<Space>: 进入下一屏。  b键: 在文本中前移  <Enter>: 向下移动一行  q键或<Ctrl+C>:退出。  cat命令适用于快速查看文件内容，但当文件很长时可能不太方便。  more命令提供了分页显示的功能，但功能相对简单。  less命令是more命令的增强版，提供了更多的导航和搜索功能，是查看长文件的理想选择。  15. 复制/etc/passwd 文件到你的主目录, 并将其重命名为usersfile.用一条命令实现。如果用两条命令实现呢？    如果用两条命令则是  cp /etc/passwd ~  mv ~/passwd ~/usersfile  **16. 用wc命令统计文件userfile的字符数、行数和单词数. 解释显示出的各栏对应的内容。如果只统计行数呢？**      17. 分别创建userfile的一个硬链接文件userfile-hard和一个字符链接文件userfile-soft。考虑userfile-hard和userfile-soft的异同点。  之后删除userfile文件，查看一下userfile-hard和userfile-soft的内容并解释结果的不同之处。      解释结果的不同之处  硬链接：由于硬链接和原文件共享相同的数据块，因此删除原文件对硬链接没有影响。硬链接仍然可以访问和显示文件的内容。  软链接：符号链接是一个指向原文件路径的文件。当原文件被删除时，符号链接变得无效，因为它指向的路径不再存在。尝试访问符号链接会导致错误，因为它现在是一个“死链接”。  18. 用whereis搜索命令passwd所在的路径。用locate命令搜索文件名中带有aa字段的文件。 whereis和locate命令的区别是什么？      whereis和locate命令的区别  搜索范围和内容：  whereis主要用于搜索命令的可执行文件、源代码文件和手册页。它搜索的是系统环境变量中指定的目录，如/usr/bin、/usr/sbin、/usr/lib等（这些目录可以通过whereis -l命令查看）。  locate则用于搜索整个文件系统中的文件名。它依赖于一个预先构建的数据库，该数据库包含了系统中所有文件的路径信息。因此，locate的搜索速度非常快，但结果可能不是实时的，因为数据库需要定期更新。  更新频率：  whereis每次运行时都会实时搜索指定的目录，因此结果总是最新的。  locate的搜索结果取决于数据库的更新频率。数据库通常由系统自动定期更新（如每天一次），但你也可以手动运行sudo updatedb命令来更新数据库。  搜索速度：  由于whereis需要实时搜索目录，因此其搜索速度可能相对较慢，特别是当搜索范围很大时。  locate则因为是在数据库中搜索，所以速度非常快，几乎可以立即返回结果。  搜索类型：  whereis主要用于搜索与命令相关的文件。  locate则可以搜索任何类型的文件，只要它们的路径信息被包含在数据库中。  三. 对文件和目录的递归操作  19. 创建一个sub1目录, 在sub1中再创建一个 sub2 目录. 用一条命令来实现.    20. 转到 sub2 目录, 创建一个文件 myfile.    21. 返回到你的主目录. 复制整个 sub1目录树到tree1. 分别递归显示目录sub1和tree1里的所有文件.      22. 现在你有了2个目录树, sub1和tree1. 把目录树tree1 移到sub1子目录中.    23. 列出你的主目录的内容. 递归显示sub1目录中的所有文件和目录.    四. 创建用户帐号  为了完整地演示权限, 我们需要创建一些附加的用户, tux1和tux2, 它们都是penguins 组的成员.  （使用虚拟机的同学可以通过如下提示的Ctrl+Alt+Fn键，打开多个字符终端，但使用云桌面无法切换，可以直接桌面点击鼠标右键-打开终端，使用多个图形界面终端代替）   1. 切换到tty3(按Ctrl+Alt+F3键),用 root帐号登录.     2. 执行下列一系列命令:  # groupadd penguins  # useradd -m -g penguins -c "Tux the Penguin (1)" tux1  # useradd -m -g penguins -c "Tux the Penguin (2)" tux2  # passwd tux1  New password: penguin1  Retype new password: penguin1  # passwd tux2  New password: penguin2  Retype new password: penguin2    3. 切换到tty1(按Ctrl+Alt+F1键), 用tux1帐号登录, 密码是penguin1, 然后再切换到 tty2(按Ctrl+Alt+F2键),  用tux2帐号登录, 密码是penguin2.    三个用户可以可以根据密码登录  五. 文件和目录的权限  24. 切换到tty1(按Ctrl+Alt+F1键), 你是以 tux1帐号登录的, 查看你的主目录的权限.    25. 切换到tty2, 你是以 tux2帐号登录的. 试着改变到tux1的主目录, 或者显示tux1主目录的内容. 能行吗?为什么?    不能，没权限  26. 切换到tty1. 改变tux1主目录的权限以使其他用户能够有读(read)和执行(execute)的权限. 然后再以tux2帐号试着进入tux1的主目录. 现在能行吗?      一开始不能访问，开放权限可以访问  在tux2用户的终端中，你可以使用ls -l命令查看tux1主目录内的文件列表，并检查它们的权限。你应该会看到，尽管tux2用户能够进入该目录，但大多数文件仍然只对tux1用户可读（和可能可写）。  27. 作为用户tux2, 试着创建和删除tux1的主目录里的文件. 能否成功?    没有权限  28. 再次切换到tty1. 创建一个bin目录,复制文件/bin/ls 到此目录,并且重命名为my\_ls.    29. 设置my\_ls的权限为rw-r-----, 然后分别以tux1和tux2登录试着运行它. 能否运行? 为什么?  不能运行，没有运行权限      30. 重新设置my\_ls的权限为rwxr-xr-x, 然后再次分别以tux1和tux2登录试着运行它. 现在能否运行?    Tux1能够运行  Tux2中也能够运行  31. 试着运行 my\_ls , 分别以tux1 , tux2, 和你自己的帐号 , 权限分别为 rw-------, rw-rw----, rwx------, rwx--x--- 和 rwx--x--x . 对于tux1, 运行my\_ls所需要的最小权限是什么? 对于tux2呢? 对于你自己的帐号呢?    **用户角色和权限设置**   * **tux1** 是文件的所有者。 * **tux2** 是文件的组成员。 * 您自己的帐号是文件的其他用户（非所有者和非组成员）。   **权限分析**   1. **rw-------**：    * tux1（所有者）：可以读和写，但不能执行。因此，tux1 无法运行 my\_ls。    * tux2（组成员）：没有权限。    * 您（其他用户）：没有权限。   结论：此权限设置下，无人可以运行 my\_ls。   1. **rw-rw----**：    * tux1（所有者）：可以读和写，但不能执行。因此，tux1 无法运行 my\_ls。    * tux2（组成员）：可以读和写，但不能执行。因此，tux2 也无法运行 my\_ls。    * 您（其他用户）：没有权限。   结论：此权限设置下，无人可以运行 my\_ls。   1. **rwx------**：    * tux1（所有者）：可以读、写和执行。因此，tux1 可以运行 my\_ls。    * tux2（组成员）：没有权限。    * 您（其他用户）：没有权限。   结论：对于 tux1 来说，最小权限是 rwx------。   1. **rwx--x---**：    * tux1（所有者）：可以读、写和执行。因此，tux1 可以运行 my\_ls。    * tux2（组成员）：没有读或写权限，但可以执行。然而，通常脚本或程序需要读权限来正确执行（例如，读取脚本内容）。但在这个特定设置中，如果 my\_ls 是一个不需要读取其他文件内容的独立可执行文件，则 tux2 可以运行它。不过，这种情况比较罕见。    * 您（其他用户）：可以执行，但没有读或写权限。同样，这通常不是一个好的做法，因为程序可能无法正确运行。   结论：这个设置对于 tux1 是足够的，但对于 tux2 来说可能不是最佳实践，因为它缺少读权限。然而，如果我们只关注执行权限，则这是 tux2 能获得执行权限的最小设置（尽管不推荐）。   1. **rwx--x--x**：    * tux1（所有者）：可以读、写和执行。    * tux2（组成员）：可以执行，但没有读或写权限。与上一个设置相同，这通常不是一个好的做法。    * 您（其他用户）：可以执行，但没有读或写权限。   结论：这个设置对于 tux1 来说仍然是足够的，但对于 tux2 和其他用户来说，它同样缺少读权限，通常不推荐。  **所需最小权限**   * 对于 **tux1** 来说，最小权限是 rwx------，因为这允许他读、写和执行文件。 * 对于 **tux2** 来说，如果 my\_ls 是一个不需要读取其他文件内容的独立可执行文件，并且你只关心执行权限而不关心读或写权限，则最小权限可以是 ---x--x--x（但通常不推荐，因为缺少读权限可能导致程序无法正确运行）。然而，在提供的选项中，最接近的是 rwx--x--- 或 rwx--x--x，其中 tux2 至少可以获得执行权限。但请注意，这些设置并不理想，因为通常程序也需要读权限。 * 对于您自己的帐号来说，如果您需要运行 my\_ls，则最小权限是文件对所有用户开放执行权限的设置，即 rwx--x--x。但同样，这通常不是一个好的安全实践，因为它允许任何用户执行该文件。 | | | |
| 结论分析与体会：  **结论分析**   1. **对目录和文件操作的理解**：    * 通过本次实验，我深入理解了Linux系统中目录和文件的基本操作，包括如何查看当前目录（pwd）、改变目录（cd）、列出目录内容（ls）、创建和删除目录及文件（mkdir, rmdir, touch, rm）、复制和移动文件（cp, mv）等。    * 我学会了使用长列表格式（ls -l）来查看文件和目录的详细信息，包括权限、所有者、大小、修改时间等。    * 掌握了符号链接（ln -s）和硬链接（ln）的创建方法，并理解了它们之间的区别。 2. **对文件和目录权限的掌握**：    * 通过umask命令，我学会了设置新创建文件和目录的默认权限。    * 使用chmod命令，我能够灵活地修改文件和目录的权限，以满足不同的安全需求。    * 通过创建用户tux1和tux2，并设置不同的权限，我深刻理解了Linux系统中权限模型的工作原理。特别是，我认识到执行权限对于文件和目录的重要性：对于文件，执行权限允许用户运行该文件；对于目录，执行权限允许用户进入该目录并访问其内容。    * 在实验过程中，我发现即使为其他用户设置了读和执行权限，如果没有写权限，他们仍然无法在该目录中创建或删除文件。这进一步强调了权限设置的细致性和重要性。 3. **对其他命令的熟悉**：    * 我学会了使用wc命令来统计文件的字符数、行数和单词数，这对于文本处理和分析非常有用。    * du和df命令帮助我了解了磁盘空间的使用情况，这对于系统管理和优化至关重要。    * 通过more、less和cat命令的比较，我理解了它们在查看文件内容时的不同特点和适用场景。   **个人心得体会**   1. **实践出真知**：    * 理论知识虽然重要，但只有通过实践才能真正掌握。在这次实验中，我通过亲手操作Linux命令，深刻理解了文件和目录操作以及权限设置的原理和方法。    * 实践过程中遇到问题时，我学会了查阅文档、搜索解决方案和请教他人，这些技能对于未来的学习和工作都非常有帮助。 2. **细节决定成败**：    * 在Linux系统中，一个小小的权限设置错误或命令错误都可能导致严重的后果。因此，我深刻认识到在操作时必须非常细心和谨慎。    * 同时，我也意识到在理解Linux系统时，不能只看表面现象，而要深入探究其背后的原理和机制。 3. **持续学习的必要性**：    * Linux系统是一个庞大而复杂的生态系统，其功能和特性不断发展和完善。因此，我认识到持续学习是保持竞争力的关键。    * 在未来的学习和工作中，我将继续深入学习Linux系统的相关知识和技术，不断提升自己的能力和水平。 | | | |

1. 当次实验结束后一周内按班组织上交实验报告。

2. 实验报告文件命名为：“学号+姓名”，格式为WORD文档。