



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

Linux 系统及应用作业报告

作业 3: 文件时间

学院 卓越学院

学号 23040447

姓名 陈文轩

专业 智能硬件与系统 (电子信息工程)

2025 年 4 月 28 日

1 问题 1

”Access Time” 中的”Access” 是指哪种操作？设计一个示例，操作演示一下 Access Time 的变化。

1.1 Access Time 定义

在 Linux 系统中，”Access Time”（访问时间，简称 atime）指的是文件最后一次被读取或执行的时间。这包括但不限于以下操作：

- 使用 cat, less, head 等命令查看文件内容
- 程序读取文件内容
- 执行脚本文件

1.2 示例演示

Code Listing 1: 演示 Access Time 变化

```
1 # 1. 创建测试文件并查看初始时间戳
2 echo "This is a test file." > testfile.txt
3 stat testfile.txt
4
5 # 2. 访问文件内容
6 cat testfile.txt
7
8 # 3. 再次查看时间戳，观察 Access Time 变化
9 stat testfile.txt
```

1.3 预期结果

- 第一次 stat 命令输出中的 Access Time 显示文件创建时间
- 执行 cat 命令后，Access Time 更新为执行 cat 时的时间
- Modify Time 和 Change Time 保持不变

1.4 实际结果

```
cc@HpVictus9:~$ echo "This is a test file." > testfile.txt
cc@HpVictus9:~$ stat testfile.txt
  File: testfile.txt
  Size: 21          Blocks: 8          IO Block: 4096   regular file
Device: 8,32      Inode: 51490         Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   cc)   Gid: ( 1000/   cc)
Access: 2025-04-28 11:06:14.242531675 +0800
Modify: 2025-04-28 11:06:14.242531675 +0800
Change: 2025-04-28 11:06:14.242531675 +0800
 Birth: 2025-04-28 11:06:14.242531675 +0800
cc@HpVictus9:~$ cat testfile.txt
This is a test file.
cc@HpVictus9:~$ stat testfile.txt
  File: testfile.txt
  Size: 21          Blocks: 8          IO Block: 4096   regular file
Device: 8,32      Inode: 51490         Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   cc)   Gid: ( 1000/   cc)
Access: 2025-04-28 11:07:03.602075623 +0800
Modify: 2025-04-28 11:06:14.242531675 +0800
Change: 2025-04-28 11:06:14.242531675 +0800
 Birth: 2025-04-28 11:06:14.242531675 +0800
cc@HpVictus9:~$
```

图 1 WSL 实际测试效果

可以发现，只有 Access Time 时间变化，Modify Time 和 Change Time 保持不变，符合预期。

2 问题 2

设计一个示例，演示”Modified Time”和”Change Time”的区别。

2.1 时间戳区别

- **Modified Time (mtime):** 文件内容最后一次被修改的时间
- **Change Time (ctime):** 文件元数据（权限、所有者等）或内容被修改的时间

2.2 示例演示

Code Listing 2: 演示 mtime 和 ctime 区别

```
1 # 1. 创建文件并记录初始时间戳
2 echo "Initial content" > demo.txt
3 stat demo.txt
4
5 # 2. 修改文件内容（会更新mtime和ctime）
6 echo "Modified content" > demo.txt
7 stat demo.txt
```

```

8
9 # 3. 仅修改文件权限（只更新 ctime）
10 chmod 600 demo.txt
11 stat demo.txt

```

2.3 预期结果

操作	Modified Time	Change Time
初始创建	T1	T1
修改内容	T2	T2
修改权限	T2	T3

其中 $T1 < T2 < T3$ ，表明：

- 修改内容会同时更新 mtime 和 ctime
- 修改权限只会更新 ctime

2.4 实际结果

```

cc@HpVictus9:~$ echo "Initial content" > demo.txt
cc@HpVictus9:~$ stat demo.txt
  File: demo.txt
  Size: 16          Blocks: 8          IO Block: 4096   regular file
Device: 8,32      Inode: 49423         Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/      cc)   Gid: ( 1000/      cc)
Access: 2025-04-28 11:04:54.730451446 +0800
Modify: 2025-04-28 11:19:19.421699628 +0800
Change: 2025-04-28 11:19:19.421699628 +0800
Birth: 2025-04-28 11:04:54.730451446 +0800
cc@HpVictus9:~$ echo "Modified content" > demo.txt
cc@HpVictus9:~$ stat demo.txt
  File: demo.txt
  Size: 17          Blocks: 8          IO Block: 4096   regular file
Device: 8,32      Inode: 49423         Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/      cc)   Gid: ( 1000/      cc)
Access: 2025-04-28 11:04:54.730451446 +0800
Modify: 2025-04-28 11:20:16.511700596 +0800
Change: 2025-04-28 11:20:16.511700596 +0800
Birth: 2025-04-28 11:04:54.730451446 +0800
cc@HpVictus9:~$ chmod 600 demo.txt
cc@HpVictus9:~$ stat demo.txt
  File: demo.txt
  Size: 17          Blocks: 8          IO Block: 4096   regular file
Device: 8,32      Inode: 49423         Links: 1
Access: (0600/-rw-----)  Uid: ( 1000/      cc)   Gid: ( 1000/      cc)
Access: 2025-04-28 11:04:54.730451446 +0800
Modify: 2025-04-28 11:20:16.511700596 +0800
Change: 2025-04-28 11:20:27.921735096 +0800
Birth: 2025-04-28 11:04:54.730451446 +0800
cc@HpVictus9:~$

```

图 2 WSL 实际测试效果

可以发现，执行第二个“echo ”Modified content”> demo.txt”（修稿文件内容指令），Modified Time 和 Change Time 都有更改。“chmod 600 demo.txt”（更改

文件权限，使文件所有者可读写，其他用户无权限）指令执行后，只有 **Change Time** 更新。