GlobalLogic
A Hitachi Group Company

EDUCATION

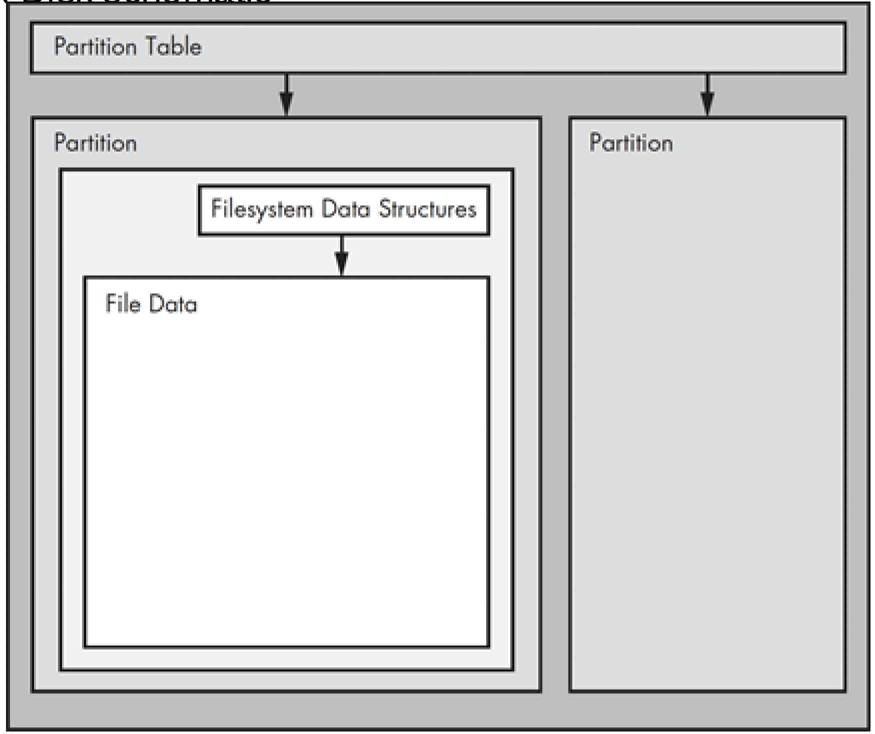
Smart Start: Linux/Networking File System

Sergii Kudriavtsev

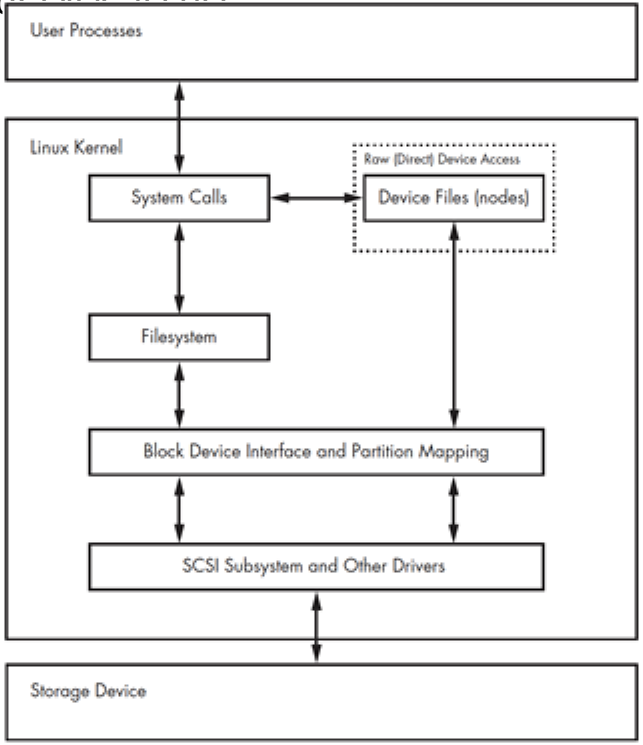
Agenda

Schematic of typical Linux Disk

- Typical Linux Disk schematic



- Kernel schematic for Disk access


User Processes
Linux Kernel
Raw (Direct) Device Access
System Calls
Device Files (nodes)
Filesystem
Block Device Interface and Partition Mapping
SCSI Subsystem and Other Drivers
Storage Device

- Disk Partitioning and partition structure
  - MBR - Master Boot Record
  - GPT - GUID Partition Table //GUID - globally unique identifier
- **Number of the partitions supported**

  MBR - 64 bytes partition table, 4 primary partitions or 3 primary + 1 extended with up to 128 sub partitions.

  GPT - 16,384 bytes partition table, up to 128 primary partitions.

- **Partition size supported**

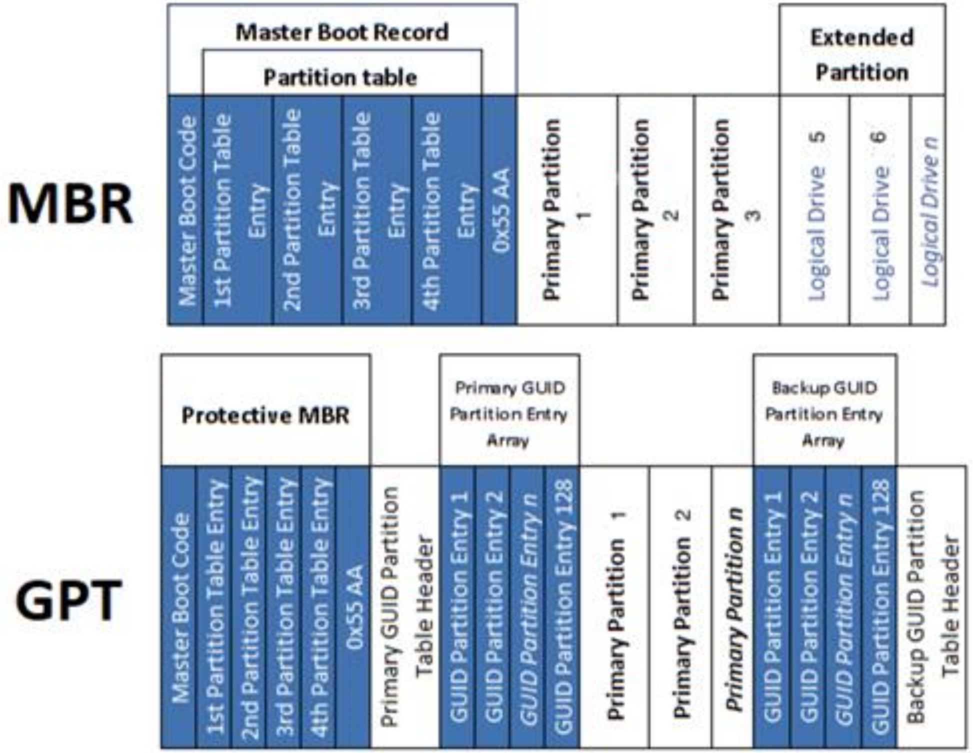
  MBR - 4 bytes (32 bit) size per partition, maximum supported disk size is 2TB

  GPT - 8 bytes (64 bit) size per partition, maximum supported disk size 9.4 ZB (+ OS limitation).

- **Redundancy**

  MBR - stores the boot and partition data in one place

  GPT - stores multiple copies of the boot and partition data across the disk + has CRC



- Partitioning DISK devices
  - parted/gparted - Supports both MBR and GPT
  - fdisk - Supports MBR only
  - gdisk - Supports GPT only

```
$ parted -l

Model: PM9B1 NVMe Samsung 512GB (nvme)
Disk /dev/nvme0n1: 512GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size     File system  Name                  Flags
 1      1049kB  538MB   537MB    fat32        EFI System Partition  boot, esp
 2      538MB   2330MB  1792MB   ext4
 3      2330MB  512GB   510GB
```

```
$ parted -l

Model: SP DS72 (scsi)
Disk /dev/sda: 250GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  1128MB  1127MB  primary  fat32        boot
 2      1128MB  3276MB  2147MB  primary  ext4
 3      3276MB  250GB   247GB   primary
```

/dev/sdX /dev/hdX and /dev/xvdX - /dev/sd$LETTER$PARTITIONNUMBER, `sd-bus/ide-scsi` driver, used for SCSI disks, SATA disks, and USB disks.

/dev/mmcblkXpY which is for SD/eMMC/bare NAND/NOR devices

/dev/nvme - NVME device connected to port and uses the nvme driver on Linux. NVMe has the concept of namespaces: /dev/nvme$LETTER**$NAMESPACE**$PARTITIONNUMBER

Filesystem and FS Types

- Filesystem
  - Filesystem - is a form of database; it supplies the structure to transform a simple block device into the sophisticated hierarchy of files and subdirectories that users can understand. Has a tree-like directory structure and I/O interface.
  - Filesystem tasks:
    - Data storage
    - System interfaces (/sys and /proc)
  - Filesystems implementation:
    - Kernel, kernel modules
    - File System in User Space (FUSE)
    - VFS - ensures that all filesystem implementations support a standard interface so that user-space applications access files and directories in the same manner

GlobalLogic
A Hitachi Group Company
EDUCATION

- FS Type
  - Linux: EXT2 (no journaling) (UFS + FFS)
  - Linux: EXT3 (+journalling), EXT4 (current, + larger files + greater subdirs number),
  - Linux: iso9660 - CD-ROM standard FS.
  - FAT filesystems (msdos, **vfat**, umsdos) - Microsoft systems FS Support.
  - HFS+ (hfsplus) is an Apple standard used on most Macintosh systems.
  - Btrfs - copy-on-write (CoW) filesystem for Linux (Future is almost here…). Developed at Oracle, Red Hat, Fujitsu, Intel, SUSE, STRATO.

  - Another native Linux FS:
    - XFS — a high-performance journaling filesystem (FS for big files).
    - ReiserFS/Raiser4 - general-purpose, journaling file system, (tail packing, reduce fragmentation, dynamic resize, e.t.c.)

  - Special-Purpose Filesystems:
    - proc - Mounted on /proc. Represent various aspects of the processes.
    - sysfs - Mounted on /sys. Pseudo file system exports information about various kernel subsystems.
    - tmpfs - Temporary File System, mounted on /run/ or /var/run/, /tmp. Volatile memory.

- FS Type
  - Filesystems for NOR and NAND flash memory: To use raw flash chips for mass storage, you have to use a filesystem that understands the peculiarities of the underlying technology. There are three such filesystems:
    - JFFS2 - Journaling Flash File System 2. Works for NOR and NAND memory.
    - YAFFS2 - Yet Another Flash File System 2. Similar to JFFS2, but specifically for NAND. Adopted by Google, Preferred for Android.
    - UBIFS - Unsorted Block Image File System. Both NOR and NAND memory, which is used in conjunction with the UBI block driver.
    - Overlayfs, Unionfs, and aufs - union filesystems, that allow multiple filesystems to be combined and presented to the user as a single tree.
    - SquashFS - compressed read-only FS.

- FS Type
  - Networking File Systems(protocols):
    - Distributed file systems - file management concept across multiple servers that communicate over a network and provide remote file sharing and access. (Mount approach).
    - NFS (Network File System) is a file-sharing protocol primarily used in Unix-like operating systems. This protocol consists of a client and an NFS server as the central repository of files and directories.
    - SMB (Server Message Block) protocol is primarily used in Windows systems. Has a user-based authentication system, commonly used for print-sharing capabilities.
    - CIFS (Common Internet File System), is an implementation of the SMB protocol. Microsoft developed CIFS solely based on the original version of SMB, called SMB1. Consequently, later versions of SMB provide more capabilities and enhancements than the CIFS implementation.

- FS Type
  - FS Type commands:
    - `$ df -T /`
    - `$ df -H /`
    - `$ mount`
    - `$ cat /proc/mounts`
  - List of block devices
    - `$ lsblk`
  - Mounting rules
    - `/etc/fstab`

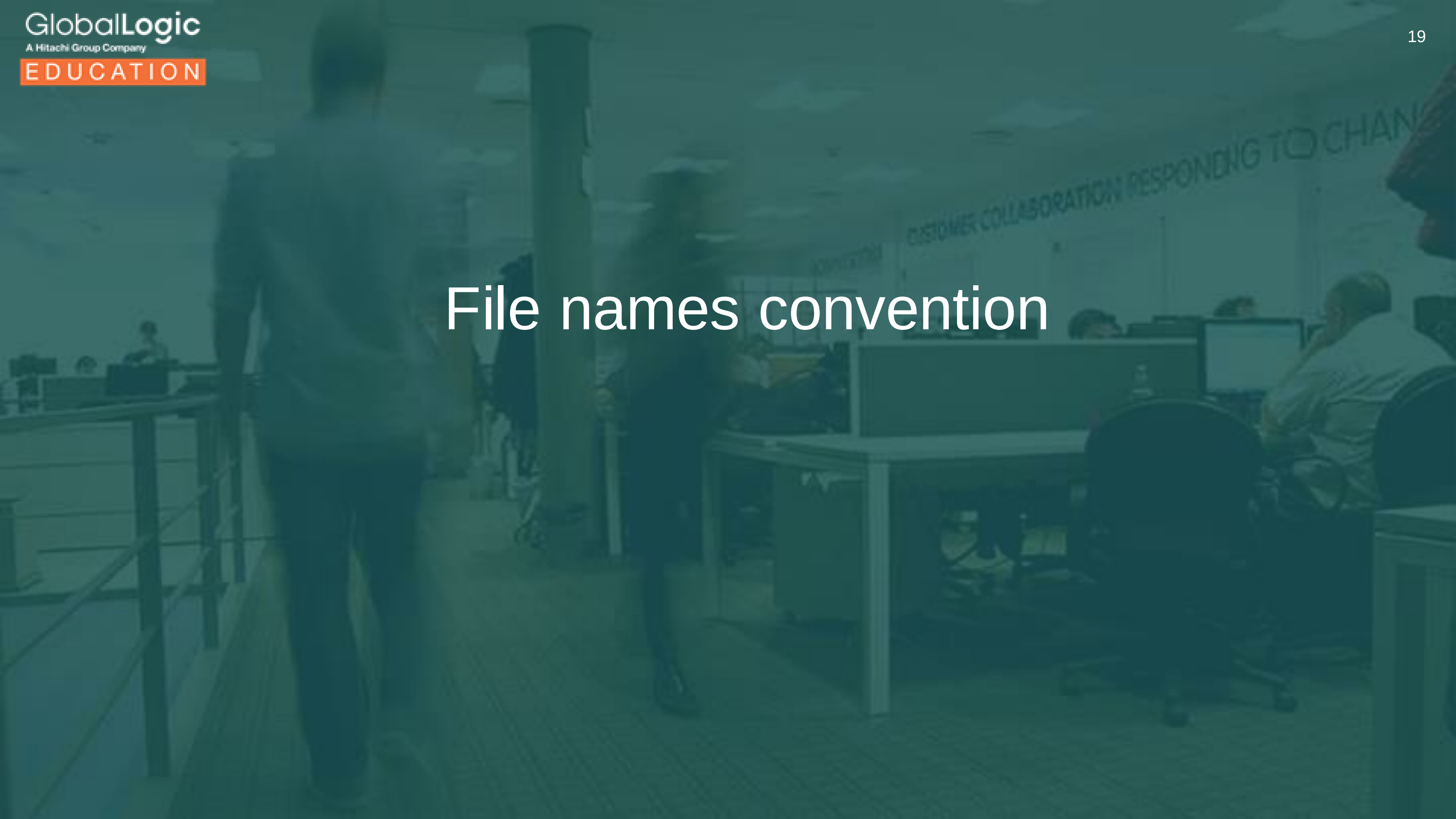
Hierarchy

- Hierarchy
  - /
  - /bin, /sbin
  - /boot
  - /etc
  - /home
    - /home/*username*
  - /opt
    - s/kernel/<*setting_name*>
      - `$ cat /proc/sys/kernel/threads-max`
      - `$ cat /proc/sys/kernel/hostname`
    - For 3PSW (third-party software)

- Hierarchy
  - /proc
    - `$ man 5 proc`
    - Kernel config
      - `$ zcat /proc/config.gz`
      - `$ zless /proc/config.gz`
    - /proc/<*PID*>/
      - `$ cat /proc/<PID>/cmdline`
    - /proc/sys
  - /root
  - /usr
    - /usr/bin, /usr/local/bin, /usr/sbin
  - /var
  - /tmp

Aliases of directories

- Aliases of directories
    - . - current directory
    - .. - parent directory
    - ~ - home directory
    - - - previous directory



File names convention

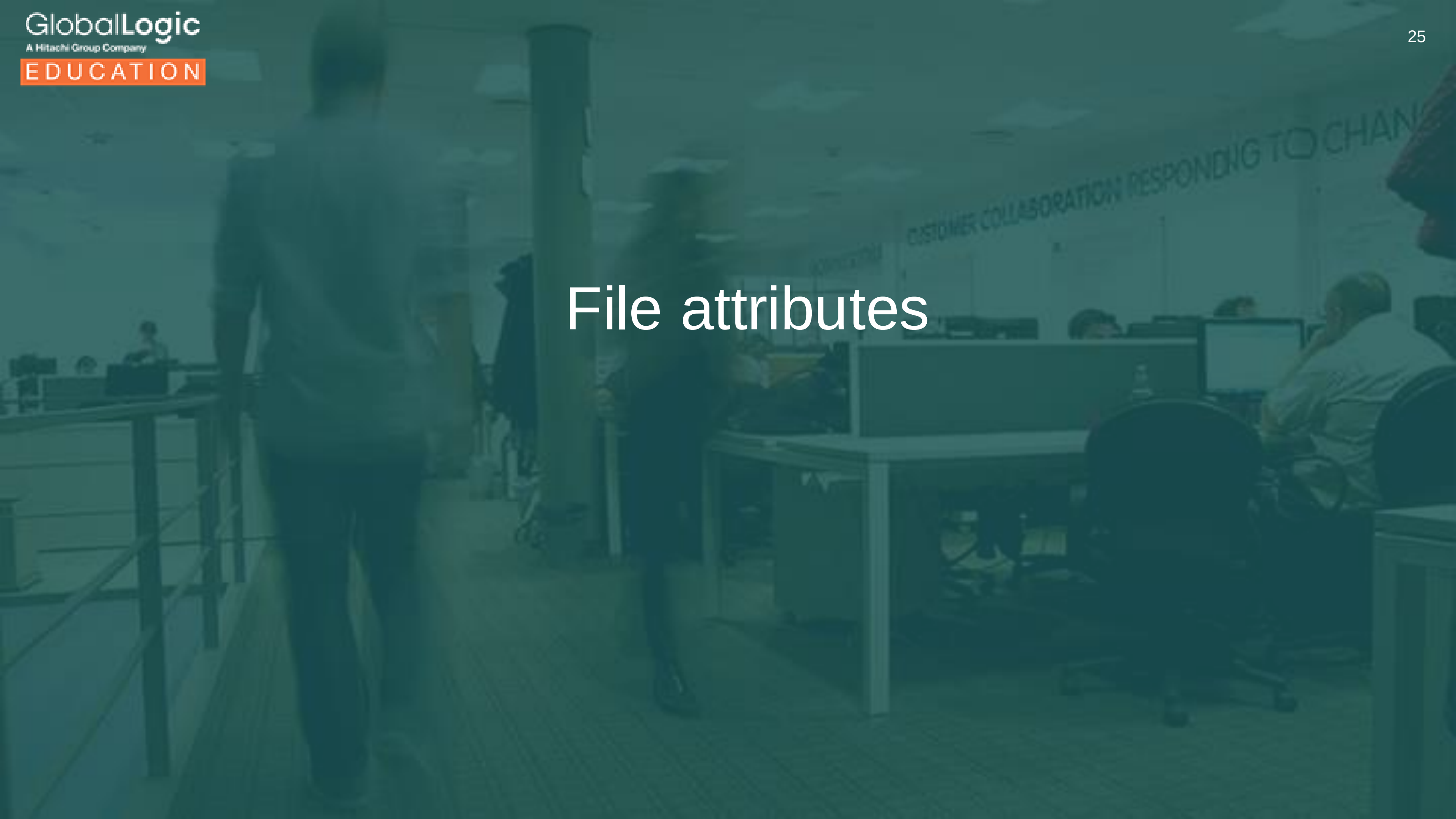
- File names convention
  - Case sensitive
  - Regular files and directories
    - `file1`
    - `File1`
  - Hidden files and directories
    - `.profile`
  - Better avoid special characters
    - Better use these characters:
      - 0-9
      - A-Z
      - a-z
      - _
      - -
      - .

Information and navigation

- $ ls # print list of files and directories under current working directory
  - /
  - $ ls *file1 file2 dir1* # list file1, file2 and contents of dir1
  - $ ls -l # use a long listing format
  - $ ls -A, $ ls -a # show also files with names starting from point: .profile, .kshrc
  - $ ls -t # sort by time of file modification: most recent files first
  - $ ls -r # sort output in reverse order
  - $ ls -R # recursive
  - $ ls -i # print inodes
  - $ ls -d # print the name of directory
  - $ ls -S # sort by file size
- $ pwd # print current working directory
- $ cd *directory* # change current working directory to *directory*
  - $ cd - # return to previous directory
  - $ cd .. # go 1 level up
  - $ cd, cd ~ # go to the home directory

- `$ find`
  - `$ find` *`directory`*
  - `$ find`
  - `$ find -name` *`file.txt`*
  - `$ find` *`directory`* `-name` *`file.txt`*
  - `$ find` *`directory`* `-name` *`'*.txt'`*
  - `$ find` *`directory`* `-name` *`'file.???'`*
  - `$ find` *`directory`* `-name` *`'file.*'`*
  - `$ find` *`directory`* `-name` *`'file.[a-z]'`*
  - `$ find` *`directory`* `-path '*bin'`
  - `$ find` *`directory`* `-type f`
  - `$ find` *`directory`* `-exec grep "`*`regexp`*`" {} \;`
  - `$ find` *`directory`* `-exec ls -l {} \;`
  - `$ find` *`directory`* `-exec ls -l {} \+`
  - `$ find` *`directory`* `-type d -or -name '*b*'`

- `$ df`
  - `$ df -k`
  - `$ df -k .`
  - `$ df -T`
  - `$ df -T .`
  - ~~`$ df -s`~~
- `$ du -s -h` *`directory`*
- `$ mount`
  - View
    - `$ mount`
    - `$ cat /proc/mounts`
  - Mount file system:
    - `$ mount` *`[OPTIONS] from_where to_where`*
    - `$ mount /dev/sda2 /mnt/my_disk`
    - `$ mount /mnt/my_disk # using rules for /mnt/my_disk directory from /etc/fstab`
- `$ lsof`
  - `$ lsof -p` *`PID`*



File attributes

- `$ ls -l`
- `$ stat file.txt`
- file type (regular file, directory, symbolic link, socket, fifo)
  - `$ file file.txt`
- modes (access rights, bitmode), rwx:
  - chmod
    - a - all
    - u - user
    - g - group
    - o - other
  - `$ chmod a+x file.txt`
    - `$ chmod +x file.txt`
  - `$ chmod g-w file.txt`
  - `$ chmod u+x file.txt`
  - `$ chmod o-r file.txt`
  - `$ chmod 755 file.txt #r - 4, w - 2, x - 1`
  - `$ chmod 0755 file.txt #r - 4, w - 2, x -`

- Special bits
  - sticky bit (for directories).
    - t equals to 1
    - Set up directory for writing by any user
      - `$ chmod +rwx directory`
      - `$ ls -ld directory`

```
drwxrwxrwx 3 user group 4096 Mar 13 12:34 directory
```

    - **Block** file deletion from this directory by other users:
      - `$ chmod +t directory`
      - `$ ls -ld directory`

```
drwxrwxrwt 3 user group 4096 Mar 13 12:34 directory
```

      - `# user 1 creates file under directory`
      - `# user 2 tries to remove file from directory and fails`
    - **Allow** file deletion from this directory by other users:
      - `$ chmod -t directory`
      - `$ ls -ld directory`

```
drwxrwxrwx 3 user group 4096 Mar 13 12:34 directory
```

      - `# user 1 creates file under directory`
      - `# user 2 tries to remove file from directory and succeeds`

- Special bits
    - s-bit (SUID, SGID) bits (for files)
        - Real user ID and group ID, effective user ID and group ID
        - Does not work for executable Shell scripts. Works for binaries.
        - `$ ls -l /usr/bin/passwd`
        - s equals to 4 for user, to 2 for group
        - Set executable bit for binary by file owner (user)

            -
              ```
              $ whoami
              user
              ```
            - `$ chmod +x /home/user/bin/test_sbit`

            -
              ```
              $ ls -l test_sbit
              -rwxr-xr-x 1 user group 7568 Mar 20 08:26 test_sbit
              ```

- Special bits
  - s-bit (SUID, SGID) bits (for files)
    - **Allow** changing effective user and group IDs for test_sbit utility:
      - Set s-bit by file owner (faust)
      - `$ whoami`

```
user
```

- `$ chmod +s /home/user/bin/test_sbit`
- `$ ls -l test_sbit`

```
-rwsr-sr-x 1 user group 7568 Mar 20 08:26 test_sbit
```

- Run the binary by other user (user2)
- `$ whoami`

```
user2
```

- `$ /home/user/bin/test_sbit`

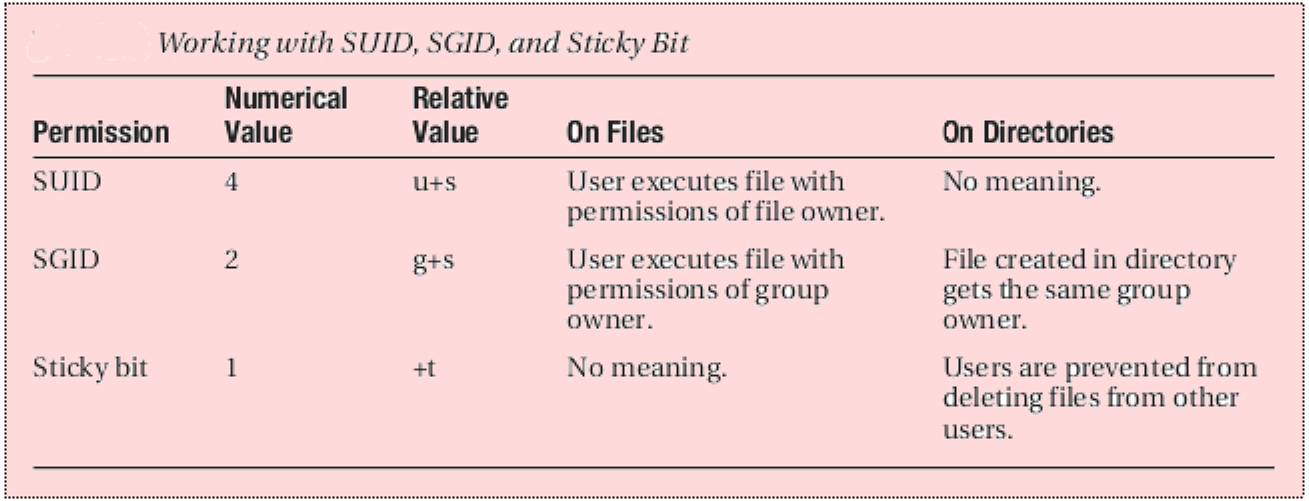
```
Real      user & group: user2, group2
Effective user & group: user, group
```

- Special bits
  - s-bit (SUID, SGID) bits (for files)
    - **Disallow** changing effective user and group IDs for test_sbit utility:
      - Remove s-bit by file owner (faust)

```
● $ whoami
user
● $ chmod -s /home/user/bin/test_sbit
● $ ls -l test_sbit
-rwxr-xr-x 1 user group 7568 Mar 20 08:26 test_sbit
```

      - Run the binary by other user (user2)

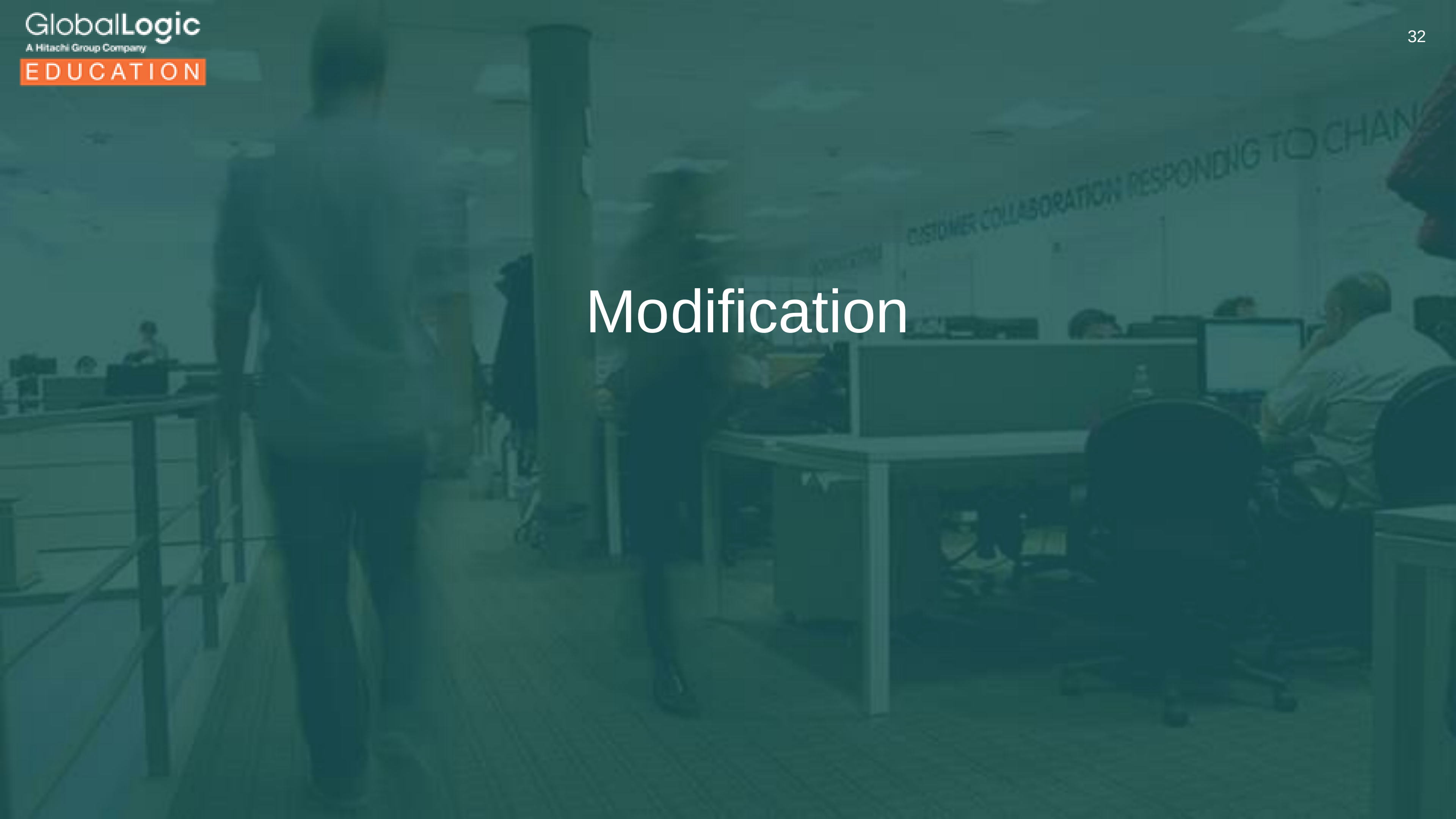
```
● $ whoami
user2
● $ /home/user/bin/test_sbit
Real      user & group: user2, group2
Effective user & group: user2, group2
```



Working with SUID, SGID, and Sticky Bit

Permission	Numerical Value	Relative Value	On Files	On Directories
SUID	4	u+s	User executes file with permissions of file owner.	No meaning.
SGID	2	g+s	User executes file with permissions of group owner.	File created in directory gets the same group owner.
Sticky bit	1	+t	No meaning.	Users are prevented from deleting files from other users.

- owner, group, other users:
  - `$ chown user file.txt # root only`
  - `$ chown :group file.txt # root only`
  - `$ chown user:group file.txt # root only`
  - `$ chown user:group directory # root only`
    - `$ chown -R user:group directory # recursively`



Modification

- Modification
  - `$ mkdir` *`directory`*
    - `$ mkdir -p` *`directory1/directory2/directory3`*
  - `$ rmdir` *`empty_directory`*
  - `$ rm` *`file.txt`*
    - `$ rm -r` *`non_empty_directory`*
    - `$ rm -f` *`non_writable_file.txt`*
    - `$ rm -i` *`file.txt`*
  - `$ touch` *`file.txt`*
    - latest modification time

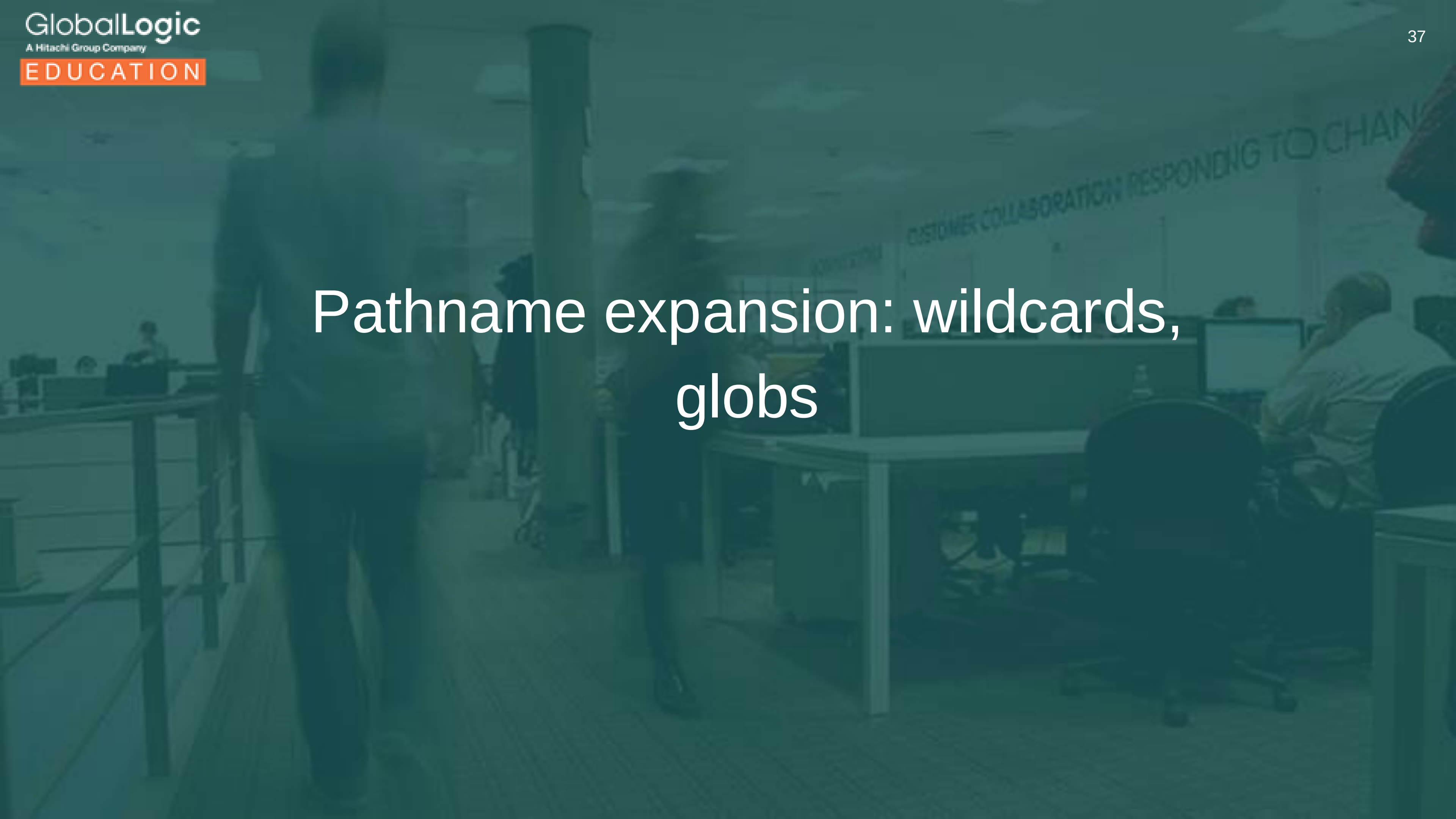
- Modification

  - `$ cp file1.txt file2.txt`
    - `$ cp file.txt directory`
    - `$ cp file.txt directory/file.txt`
    - `$ cp file1.txt directory/file2.txt`
    - `$ cp file1.txt file2.txt file3.txt file4.txt directory`
    - `$ cp -r directory1 directory2`
    - `$ cp -f file1.txt file2.txt`
    - `$ cp -i file1.txt file2.txt`

  - `$ mv file1.txt file2.txt`
    - `$ mv /home/user/file1.txt /mnt/usb_drive/file2.txt`
    - `$ mv -i file1.txt file2.txt # get a confirmation`
    - `$ mv file.txt directory`
    - `$ mv file1.txt directory/file2.txt`
    - `$ mv file1.txt file2.txt file3.txt file4.txt directory`
    - `$ mv directory1 directory2`

Links

- Links
  - Symbolic (soft)
    - `$ ln -s` *`destination link`* `# make a symbolic link:` *`link`* `->` *`destination`*
    - `$ ln -s /home/user/file1.txt /mnt/usb_drive/file2.txt`
    - `$ ls -l # see where symbolic link points to`
    - `$ readlink` *`link`*
      - `$ readlink -e` *`link`*
    - `$ pwd -P # print physical path`
  - hard
    - `$ ln` *`destination link`* `# make a hard link "link" to "`*`destination`*`"`
    - `$ ls -li # see inodes`



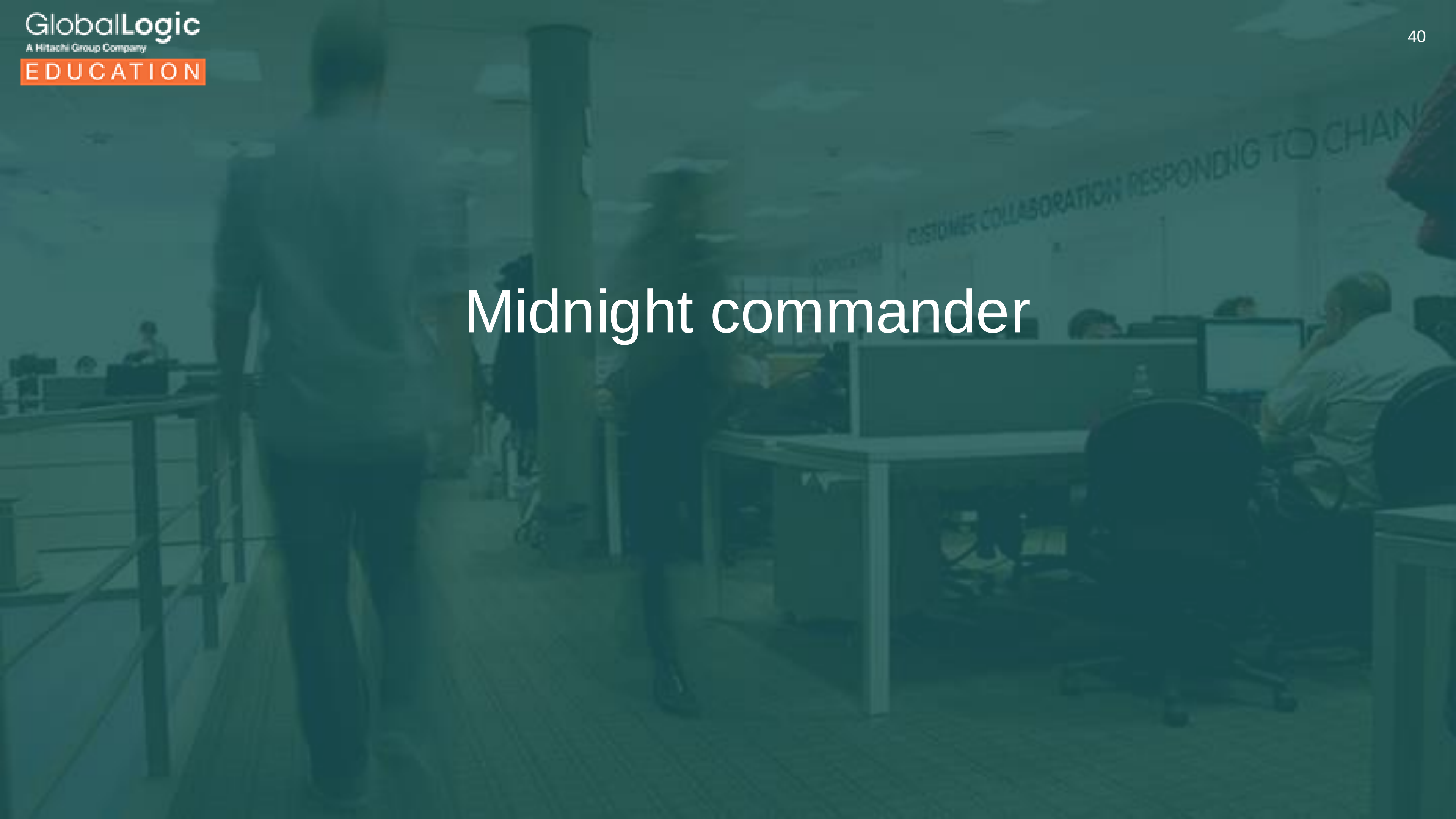
Pathname expansion: wildcards, globs

- Pathname expansion: wildcards, globs

  - https://en.wikipedia.org/wiki/Glob_(programming)
    - https://en.wikipedia.org/wiki/Glob_(programming)#Unix
  - * - matches any string, including null string
    - `$ ls *.txt # file1.txt, file2.txt, file3.txt will match`
  - ? - matches any single character
    - `$ ls file?.txt # file1.txt, file2.txt, file3.txt will match`
  - [*abcde*], [*a-e*], [*abcdek-m*], [*a-ek-m*] - matches any one of the enclosed characters
    - `$ ls file[123].txt # file1.txt, file2.txt, file3.txt will match`
    - `$ ls file[1-3].txt # file1.txt, file2.txt, file3.txt will match`
    - `$ ls file[13].txt # file1.txt, file3.txt will match`
    - `$ ls file[A-Za-z0-9].txt # fileX, filex, file1.txt, file2.txt, file3.txt will match`
    - `$ ls test_sbit.[A-Z]`

- Pathname expansion: wildcards, globs

  - [!*abcde*], [!*a-e*], [!*abcdek-m*], [!*a-ek-m*] - matches any one of the NOT ENCLOSED characters
  - `shopt`
    - `$ shopt -s` *`option_name`*
    - `$ shopt -u` *`option_name`*
    - `nocaseglob` – controls case sensitivity for path expansion
    - `dotglob` – controls path expansion for hidden files
  - For pathname expansions better use C locale to get predictable results:
    - Check current locale
      - `$ locale`
    - Set collate
      - `$ unset LC_ALL; export LC_COLLATE=C`
      - `$ unset LC_ALL; export LC_COLLATE=C.UTF-8`

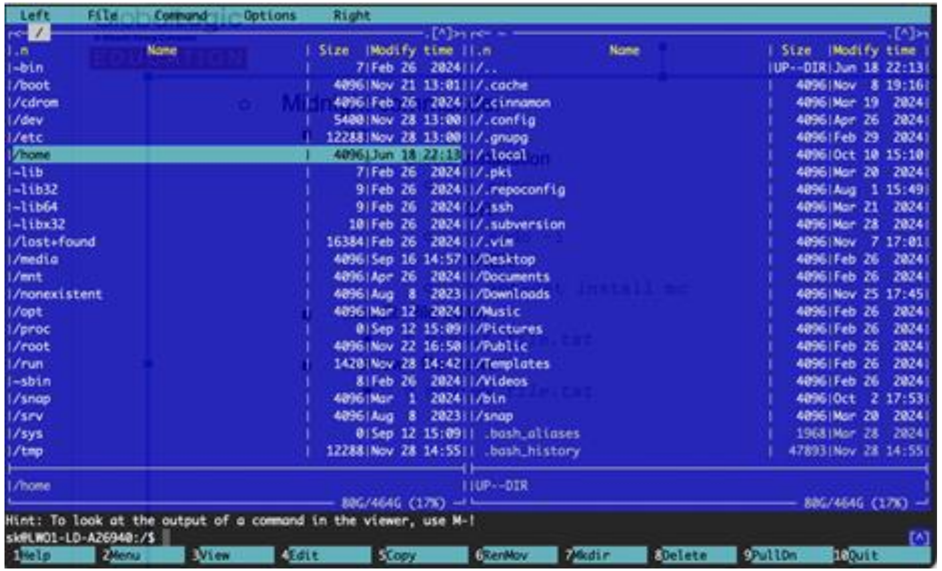


Midnight commander

- Midnight commander
  - mc
    - Global installation
      - `$ su -`

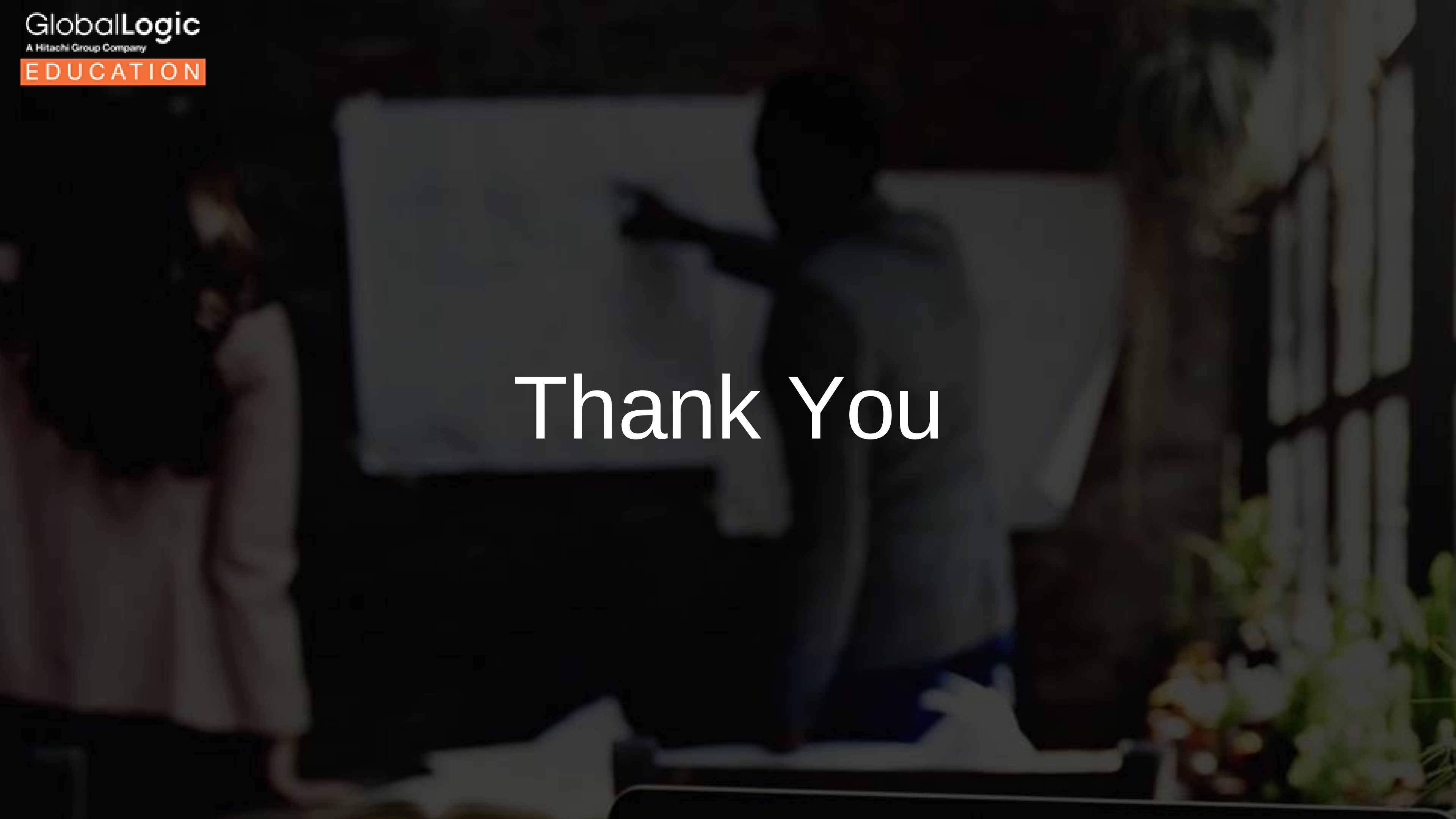
        OR
      - `$ sudo -i`

        THEN
      - `$ apt-get install mc`
  - Settings
    - `$ export EDITOR=`which mcedit``
  - mcedit - file editor
    - `$ mcedit` *`file.txt`*
  - mcview - file viewer
    - `$ mcview` *`file.txt`*



```
  Left     File     Command     Options     Right
┌<─ / ──────────────────────────────.[^]>┐┌<─ ~ ──────────────────────────────.[^]>┐
│.n           Name          │ Size  │Modify time ││.n           Name          │ Size  │Modify time │
│~bin                       │      7│Feb 26  2024││/..                        │UP--DIR│Jun 18 22:13│
│/boot                      │   4096│Nov 21 13:01││/.cache                    │   4096│Nov  8 19:16│
│/cdrom                     │   4096│Feb 26  2024││/.cinnamon                 │   4096│Mar 19  2024│
│/dev                       │   5400│Nov 28 13:00││/.config                   │   4096│Apr 26  2024│
│/etc                       │  12288│Nov 28 13:00││/.gnupg                    │   4096│Feb 29  2024│
│/home                      │   4096│Jun 18 22:13││/.local                    │   4096│Oct 10 15:10│
│~lib                       │      7│Feb 26  2024││/.pki                      │   4096│Mar 20  2024│
│~lib32                     │      9│Feb 26  2024││/.repoconfig               │   4096│Aug  1 15:49│
│~lib64                     │      9│Feb 26  2024││/.ssh                      │   4096│Mar 21  2024│
│~libx32                    │     10│Feb 26  2024││/.subversion               │   4096│Mar 28  2024│
│/lost+found                │  16384│Feb 26  2024││/.vim                      │   4096│Nov  7 17:01│
│/media                     │   4096│Sep 16 14:57││/Desktop                   │   4096│Feb 26  2024│
│/mnt                       │   4096│Apr 26  2024││/Documents                 │   4096│Feb 26  2024│
│/nonexistent               │   4096│Aug  8  2023││/Downloads                 │   4096│Nov 25 17:45│
│/opt                       │   4096│Mar 12  2024││/Music                     │   4096│Feb 26  2024│
│/proc                      │      0│Sep 12 15:09││/Pictures                  │   4096│Feb 26  2024│
│/root                      │   4096│Nov 22 16:50││/Public                    │   4096│Feb 26  2024│
│/run                       │   1420│Nov 28 14:42││/Templates                 │   4096│Feb 26  2024│
│~sbin                      │      8│Feb 26  2024││/Videos                    │   4096│Feb 26  2024│
│/snap                      │   4096│Mar  1  2024││/bin                       │   4096│Oct  2 17:53│
│/srv                       │   4096│Aug  8  2023││/snap                      │   4096│Mar 20  2024│
│/sys                       │      0│Sep 12 15:09││ .bash_aliases             │   1968│Mar 28  2024│
│/tmp                       │  12288│Nov 28 14:55││ .bash_history             │  47893│Nov 28 14:55│
├───────────────────────────────────────────────┤├───────────────────────────────────────────────┤
│/home                                          ││UP--DIR                                        │
└──────────────────────────── 80G/464G (17%) ─┘└──────────────────────────── 80G/464G (17%) ─┘
Hint: To look at the output of a command in the viewer, use M-!
sk@LWO1-LD-A26940:/$                                                                        [^]
 1Help     2Menu     3View     4Edit     5Copy     6RenMov   7Mkdir    8Delete   9PullDn  10Quit
```

GlobalLogic
A Hitachi Group Company
EDUCATION



Thank You