澳門理工學院
Instituto Politécnico de Macau
Macao Polytechnic Institute

# COMP 225 Network and System Administration

## Notes #5: File System, Storage and Backup

K. L. Eddie Law, PhD

Macao Polytechnic Institute
School of Applied Sciences
Academic Year 2020-2021, 2nd Semester

## Topics

- Understanding basic disk systems designs
- Managing server storage system and backup are vital
- Associated basic Linux file systems briefly discussed
- Types of backups – pros and cons
- Practicing backup and restore in Linux
  - Archival utility: tar
  - Task scheduler: crontab

# Basic Storage Systems

- Magnetic drive
  - Disk specifications
    - Dimensions
    - Capacity
    - RPM (revolutions per minute)
    - IOPS (input/output per second)
    - Seek time and latency
    - Hot swappable
- SSD (solid state drive)
  - No moving parts
  - Quiet, faster, and more durable

# Storage Technologies

- DAS: direct-attached storage, traditional hard drives
- NAS: network-attached storage
- SAN: storage area network (high speed sophisticated independent network)
- JBOD: just a bunch of disks (to form one large volume)

# Capacity Planning Considerations

- OS growth
  - Patches
  - Service packs
  - Log files
  - Temp files
- Data growth
  - Customer data
  - Archived data
  - Recovery data

# Mitigation Strategies

- Disk quotas
  - Soft quotas – users get alerts
- Compression
  - Loss of performance
  - For backups and archived data
- Regular cleanup
- Routine archival

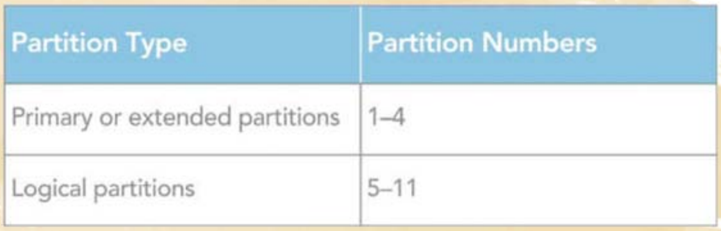
# Compress – Uncompress

- {`gzip`, `gunzip`} filename(.gz)
- {`bzip2`, `bunzip2`} filename(.bz2)
- {`xz`, `unxz`} filename(.xz)
- `zip` filename.zip filename, and `unzip` filename.zip

# Legacy System

- Boot with a BIOS
- Master boot record (MBR)
- Four real primary partitions
- Accessible size up to 2 TB
- One extended partition permitted
- Logical partitions in extended partition

- MBR partition numbering



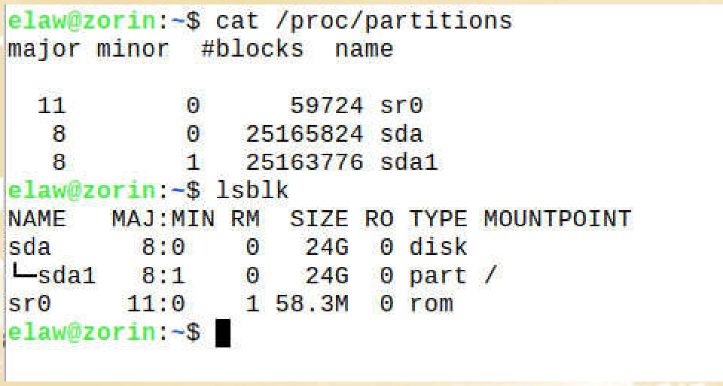
| Partition Type | Partition Numbers |
|---|---|
| Primary or extended partitions | 1–4 |
| Logical partitions | 5–11 |

# Modern Systems

- Boot with a Unified Extensible Firmware Interface (UEFI)
- Use GUID (Globally Unique Identifier) Partition Tables (GPTs)
- Unlimited number of partitions (commercial products may offer up to 128 in general)
- For current implementation, e.g., for Microsoft, the accessible size of 18 exabytes partition (1 EB = $10^{18}$ bytes)
- The theoretical upper limit is 9.4 zettabytes partition (1 ZB = $10^{21}$ bytes)

# Checking Hard Drives

- Hard drives are block devices



```
elaw@zorin:~$ cat /proc/partitions
major minor  #blocks  name

  11        0      59724 sr0
   8        0   25165824 sda
   8        1   25163776 sda1
elaw@zorin:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   24G  0 disk
└─sda1   8:1    0   24G  0 part /
sr0     11:0    1 58.3M  0 rom
elaw@zorin:~$ 
```

- Can also try out

```
$ df –h
$ sudo fdisk -l
```

# Commands – Partition Tools

- Legacy systems
  - ***fdisk***
- Example, to list partitions
  - `$ sudo fdisk -l`
  - `$ sudo fdisk -l /dev/sda`

- Modern systems
  - ***gdisk***
  - ***parted***
- Run `gdisk`
  - `$ sudo gdisk /dev/sda`

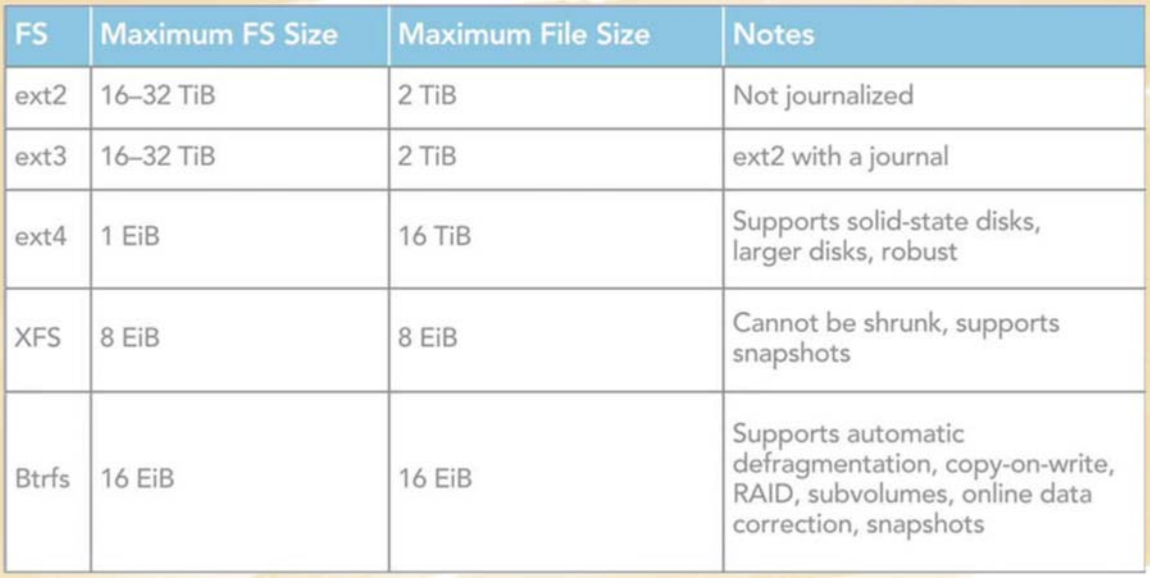
# Typical Linux File System

(Remark: logical volume manager (LVM) not discussed)

- Typical Linux file system formatting: Ext2/3/4
- Advantages with ext4
  - Improved file system size
  - Larger number of files
  - Support for solid-state disks
  - Journal checksumming
- Reformat the entire drive, e.g.,

```
$ sudo mkfs -t ext4 /dev/sda1
$ sudo lsblk -f
```

# Typical Linux File Systems



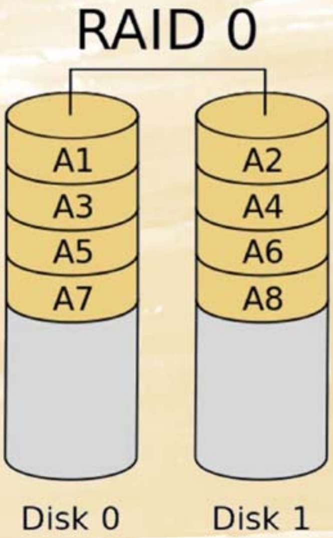
| FS | Maximum FS Size | Maximum File Size | Notes |
|---|---|---|---|
| ext2 | 16–32 TiB | 2 TiB | Not journalized |
| ext3 | 16–32 TiB | 2 TiB | ext2 with a journal |
| ext4 | 1 EiB | 16 TiB | Supports solid-state disks, larger disks, robust |
| XFS | 8 EiB | 8 EiB | Cannot be shrunk, supports snapshots |
| Btrfs | 16 EiB | 16 EiB | Supports automatic defragmentation, copy-on-write, RAID, subvolumes, online data correction, snapshots |

# RAID

Redundant Array of Independent Disks

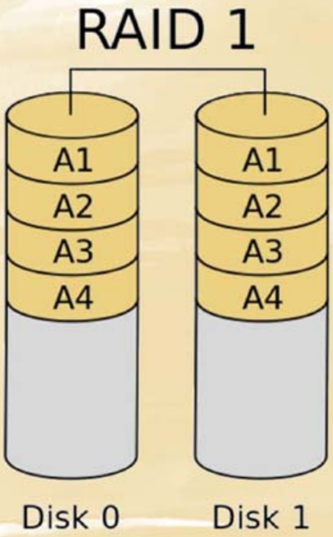
# RAID 0

- Disk striping
- No fault tolerance
- Minimum two disks
- Increased read performance
- 100% drive space utilization

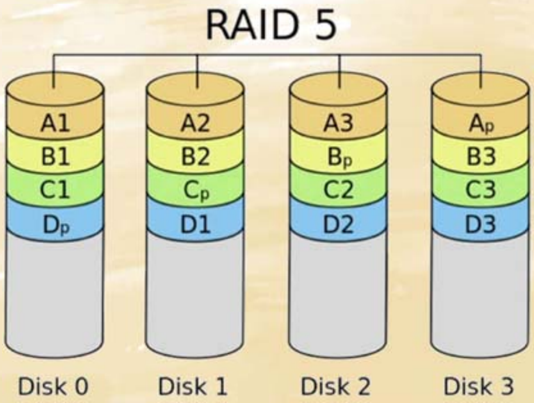


# RAID 1

- Disk mirroring
- Exactly two disks
- Increased read performance
- 50% drive space utilization
- Provides fault tolerance in of single-drive failure



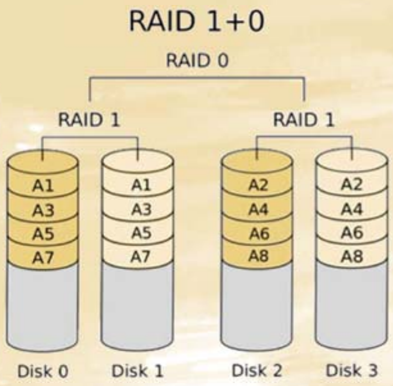
# RAID 5

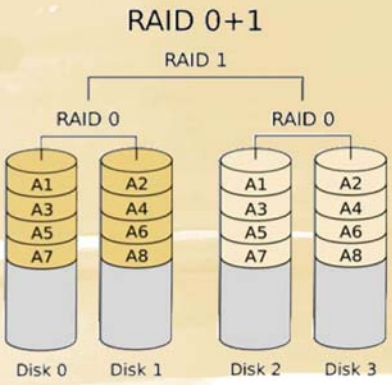


- Disk striping with parity
- Minimum three disks
- Increased read performance
- Efficient disk space utilization
  - 75% disk space utilization if 4 disks are used
  - 90% disk space utilization if 10 disks are used
- Provides fault tolerance in case of single-drive failure

# Putting RAIDs Together

- RAID 1+0 (or RAID 10)
  - Two or more mirrors that are striped
- RAID 0+1 (or RAID 01)
  - Two stripes that are mirrored





# Disasters and Backup

## Disaster Recovery

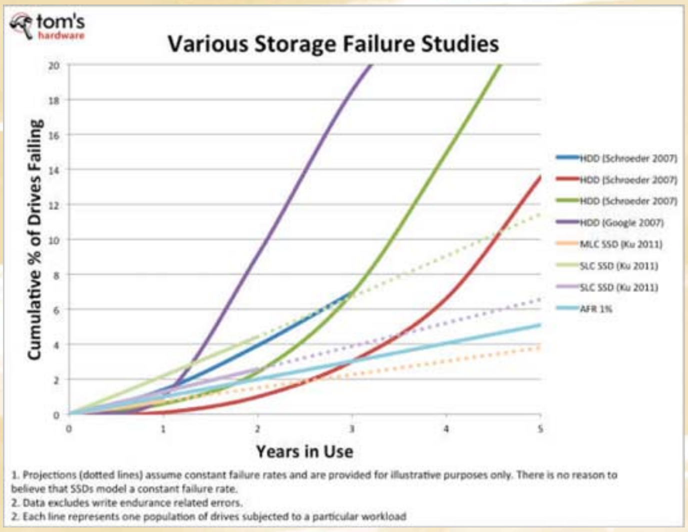
- Business continuity plan
- Disaster recovery plan
- Business impact analysis

# Replication Methods

- Disk to disk (disk mirroring)
- Server to server (failover clustering)
- Site to site

- Types of sites
  - Hot site
  - Cold site
  - Warm site

# Why Backing up Files?



1. Projections (dotted lines) assume constant failure rates and are provided for illustrative purposes only. There is no reason to believe that SSDs model a constant failure rate.
2. Data excludes write endurance related errors.
2. Each line represents one population of drives subjected to a particular workload

- With “backup and restore” operations, can protect data from loss
- Backup offers more than one copy of system files
- Purpose of restoration is to recover data that is temporarily unavailable due to some unexpected event

# Is It Good to Do Backup?

- Backup is not free
- No backup is risky
- Nowadays, in fact, cost of backing up is diminishing, value of data is growing

# Factors on Making Backups

- Determine which data is critical
  - Recovery point objective (RPO) – How much data can be lost? An hour, a week of data?
- Determine frequency and types of backups to be used
  - Recovery time objective (RTO) – How fast should data be recovered? Can we continue to operate without recovering data for a day, a week…?
- Determine categories of data, and schedule the backups accordingly

# Factors on Making Backups (cont'd)

- Determine which data is static and which is dynamic
  - Some OS installations are changed infrequently, i.e., few backups required
  - Some applications may require continuous backups, e.g., online market places
  - Understand the changing state of your client's data to determine an appropriate backup schedule
- Determine appropriate backup storage media:
  - CD/DVD
  - Tape
  - Hard Disk
  - Online

# On File System Backup

- Backing up user and system files on a single-user Linux system is a good routine operation of an ordinary user
- For complex multi-user systems, it is ***a necessary procedure*** for anyone responsible for the administration of the system
- As a system administrator, an easy-to-remember set of considerations is in the form of ***How***, ***What***, ***Why***, ***When***, ***Where***, and ***Who***?

# Questions on System Backup

- *“**How**”* – the commands, utilities, applications, or combination of hardware and software to accomplish the backup and archive; the strategies such as incrementally, in a rolling fashion, or across the entire file system structure totally, etc.
- *“**What**”* – the selected data for backup, such as user files, user account files, certain kinds of documents, the whole disk drive, multiple disk drives, subset or all of system files, etc.
- *“**Why**”* – the decisions on the relative importance of *“What”* for backing up

# Questions on System Backup (cont’d)

- *“**When**”* – how often to backup, e.g., hourly, daily, once a week, once a month, or at what time to save a particular file, etc.
- *“**Where**”* – the locations of the backup data, e.g., local disk, Cloud storage, a USB thumb drive, another computer or Network Attached Storage (NAS), etc. manually or automatically, etc.
- *“**Who**”* – the backup is carried out by a person, an automated software, or an automated process through the Cloud vendors, etc.

# Proper Backup Procedure

- Choose your application
- Scheduling
- Implementation
- Inventory (content and media)
- Verify
- Automate
- Secure

# Backup Choices

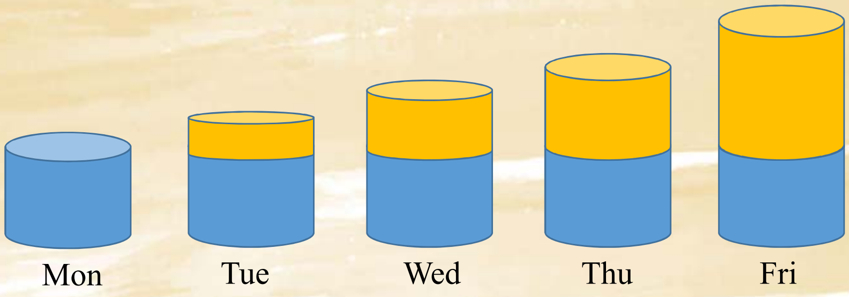
- **Full or normal backup** – all data is backed up and the ***archive bit*** is reset
- **Copy backup** – all data is backed up, but the *archive bit* is not reset
- **Incremental backup** – all data that has been changed since the last full or incremental backup is backed up, and the ***archive bit*** is reset
- **Differential backup** – all data that has been changed since the last full backup is backed up, and the ***archive bit*** is not reset
- Among file's attributes (or metadata), one bit called "***archive flag***"
  - This flag informs the backup program about which files need backing up
  - 0 for has been backed up recently; 1 for needs to be backed up

# Full Backup

- Pros
  - Provides a complete copy of all files
  - Easy to manage
  - Done less frequently than other types of backups due to cost and resource requirements, e.g., monthly, quarterly, semi-annually, annually
- Cons
  - Usually requires more media space than either differential or incremental
  - Takes a long duration to recover the full backup to a new disk

# Differential Backup

- Copy modified files since the last ***full backup***
- Differential backups grow with time, they can eventually grow larger than the last full backup
- Scheduled more frequently than a full backups: Weekly, monthly

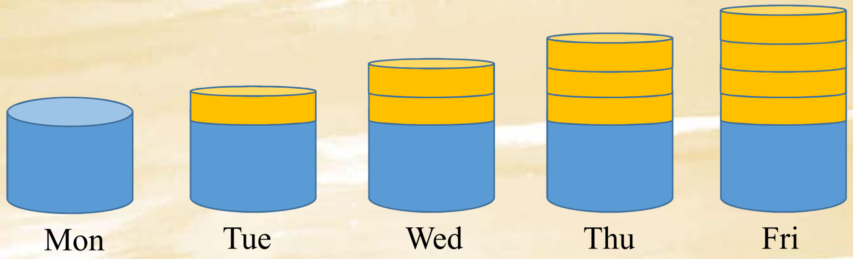


# Differential Backup (cont'd)

- Pros
  - Redundancy
  - In general, takes up less time and space than a full backup
  - If the differential backup grows to the size of the last full backup, then schedule a new full backup
- Cons
  - Redundancy – potentially many unneeded copies of the same data
  - Subsequent differentials take longer and use more media space

# Incremental Backups

- A backup of what has changed since using any type of backup the last time
- Frequency of incremental backups depends on the client needs: weekly, daily, hourly, continuously



# Incremental Backups (cont'd)

- Pros
  - Keeps a revision history of actively changing files
  - Fastest backup type
  - Uses the least amount of media to complete a single backup
- Cons
  - Much more difficult to manage
  - Failure in the chain of backup?

# Backup Strategies

- Full backups only
  - Slowest backup, fastest recovery
- Full and incremental backups
  - Once a week a full backup, each weekday an incremental backup
  - Quick backup daily, could be lengthy recovery
- Full and differential backups
  - Once a week a full backup, each weekday a differential backups
  - Backup and recovery times in-between the other two

# Example of Scheduling

- Full backup twice per year
- Differential each first Saturday morning of each month that is not scheduled for a full backup
- Incremental each Saturday morning that is not scheduled for a Full or Differential

# Backup Considerations

- Backups slow down service
  - Files should be write-locked during backup
- Avoid doing backups during peak service hours
- Schedule during early AM hours on the weekend and holidays

- Other schedule considerations
  - Consider completing a backup in conjunction with and before any major system changes are scheduled

# Backup Automation

- Automation reduces human errors
- Many pre-packaged applications include automatic scheduling
- Linux/Unix backup scripts can be submitted using the `cron` utility
- Logs can be kept in `/var/log`, and e-mail can be sent to the admin

# Data Compression

- Risks – if the media is damaged, recovery may be difficult or impossible
- Lossy
  - Some data tolerates degradation (loss of information)
- No-loss
  - Some data should not be compressed. Know your data!

# Restoration of Data

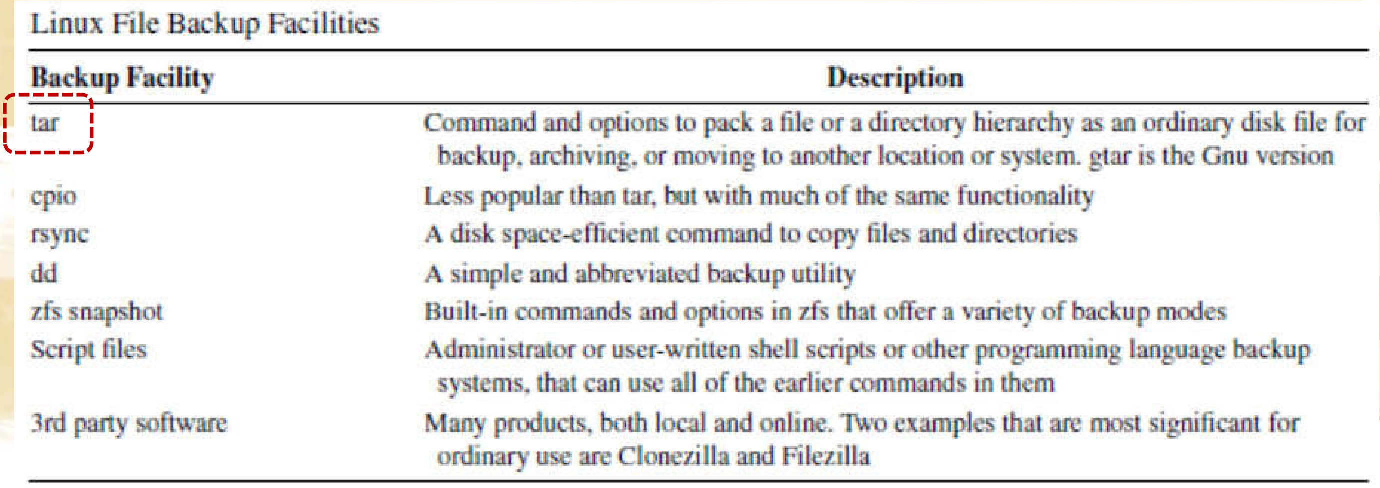
- Common reasons for restores
  - Accidental file deletion
  - Disk failure
  - Disaster recovery: fire, flood, earthquake, hacker attack, sabotage, terrorist attack, etc.

# Files and Times

- Three different times for each file (in Linux)
  - *mtime* - modification time; this value is changed when the content of the file is changed
  - *atime* - access time; the value of this is changed when the file is accessed. The atime can change when a backup utility or script read the file as well as when a use reads the file
  - *ctime* - change time; the value is updated whenever the attributes of the file change. This can be ownership or permissions
- Note: file system backups change *atime* while raw device backups will not. If implementing incremental or differential backups, this is important

# Choose your Backup Tools: Examples

- Many commercial apps are available for different OSes
- Linux/Unix



Linux File Backup Facilities

| Backup Facility | Description |
|---|---|
| tar | Command and options to pack a file or a directory hierarchy as an ordinary disk file for backup, archiving, or moving to another location or system. gtar is the Gnu version |
| cpio | Less popular than tar, but with much of the same functionality |
| rsync | A disk space-efficient command to copy files and directories |
| dd | A simple and abbreviated backup utility |
| zfs snapshot | Built-in commands and options in zfs that offer a variety of backup modes |
| Script files | Administrator or user-written shell scripts or other programming language backup systems, that can use all of the earlier commands in them |
| 3rd party software | Many products, both local and online. Two examples that are most significant for ordinary use are Clonezilla and Filezilla |

# `tar` - Linux Backup Utility

- `tar` (tape archiver) is a powerful backup and restore utility
- Most Linux files are downloaded as compressed .tar files

# Common Options for `tar`

Full optional operation name

The short form

|  |  |
|---|---|
| `-c --create` | creates a new archive |
| `-v --verbose` | lists the files being processed |
| `-x --extract` | extracts/restores the archived file |
| `-r --append` | adds the single file or directory to the archive |
| `-p –preserve-permissions` | extracts information about file permissions |
| `-f --file` | specifies the name of archive file or device location |
| `--acls` | enables POSIX ACLs that the directory has |
| `--xattrs` | enables extended attributes support |

# Data Compression for `tar`

- There are different compression algorithms, the popular ones are
  - -z --gzip --gunzip : compress or decompress using gzip function
  - -j --bzip2 : compress or decompress using bzip2 function
  - -J --xz : compress or decompress using xz function

- Keeping all original attributes (including security setup in Ses//Linux)
  - --xattrs

# Examples

- The simplest command to create an archive file from a directory

```
$ tar --create --verbose --file archive_name.tar
directory_name
```

- Backup an entire computer

```
$ sudo tar -cvpzf backup.tar.gz --exclude=/mnt /
```

- Backup up content of a web site excluding the video files

```
$ sudo tar -cvpzf wwwbackup.tar.gz
--exclude=/var/www/video /var/www
```

- Restore files

```
$ sudo tar -xvpzf wwwbackup.tar.gz -C /recover
```

Change to directory

# Job Scheduling with crontab

- Many administrative tasks must be done frequently and regularly
  - Rotating log files, backing up data
- In Linux, running jobs at regular intervals is managed by cron facility
  - Consists of the crond daemon and a set of tables describing what work is to be done and with what frequency
  - The daemon wakes up every minute and checks the crontabs to determine what needs to be done
  - The crond daemon is usually started by the init process at system startup
- Use the crontab command

# Job Scheduling

- How often to do a job?
- Many administrative tasks must be done frequently and regularly, e.g.,
  - Rotating log files, backing up data
- In Linux, running jobs at regular intervals can be managed with the `cron` (Ubuntu) or `crond` (Red Hat) facility
  - A `crond` daemon and a set of tables describing what work to do and its frequency
  - The daemon wakes up every minute and checks the `crontab`s to check what to do
  - The `crond` daemon starts when the system boots and runs as long as the system is up

# Use the `crontab` Command

- A `cron` configuration file is "`crontab`" which we call it ***cron table***
- Containing lists of command lines and times at which they are to be invoked
- `Crontab`s for individual users are stored under `/var/spool/cron` (Linux) or `/var/cron/tabs` (FreeBSD)
- There is at most one `crontab` file per user

# Editing `crontab`

- To create or edit a `crontab`, use the `crontab` command with the option `-e` (for "edit")

```
$ sudo crontab -e
```

- We can select our editor of choice using the command

```
$ sudo select-editor
```

  Or it may ask you upon creating a `cron` table at the first time

- Each `crontab` entry contains six fields
  - Comments are introduced with a pound sign (#) in the first column of a line
  - Each non-comment line contains ***six fields*** and represents one command

***minute hour dom month weekday command***

# The Fields

1. Minute of the hour (0-59)
2. Hour of the day (0-23)
3. Day of the month (dom) (1-31)
4. Month of the year (1-12)
5. Day of the week (0-6 for Sun, Mon, Tue, Wed, Thu, Fri, Sat)
6. String to be executed by bash

- For each time-related field
  - ***A star***, which matches everything
  - ***A single integer***, which matches exactly
  - ***Two integers separated by a dash***, matching a range of values
  - ***A range followed by a slash and a step value***, e.g., 1-10/2
  - ***A comma-separate list of integers or ranges***, matching any value

# Examples

- `45  10  *  * 1-5  [a bash command]`
  - “10:45 a.m., Monday through Friday.”

- `0,20,40 22-23 * 7 Fri-Sat /home/ian/mycrontest.sh`
  - Runs `mycrontest.sh` shell script at 10 pm, 10:20 pm, 10:40 pm, 11 pm, 11:20 pm, 11:40 pm, in July on Fridays and Saturdays

# Automatic Backup

- Combining `crontab` and `tar` to make an automatic backup
- Back up www web site files every minute (crazy!)

```
* * * * * sudo tar –cvpzf /backupfolder/wwwbackup.tar.gz
/var/www
```

- Back up www web site files at 3 am on Tuesdays

```
0 3 * * 2 sudo tar –cvpzf /backupfolder/wwwbackup.tar.gz
/var/www
```

# Remarks

- Discussed backup and “crontab”
- Run command “`at`” is for scheduling single event
- Logical volume group – good for scalable file systems (not discussed though)