Terminal types.

Clock:
clock also called timers are essential to the operation of any multiprogrammed system for variety of reasons.
  - maintain time of day
  - prevent one process from monopolizing the CPU among other things.
  - clock software can take the form of device driver, but it is neither a block device like disk
    neither a character like mouse.

Two clock:
the simpler clock is tied to 110 or 220 volt power line and causes and interrupt on every voltage cycle
at 50 or 60hz.
the other kind of clock built of 3 components a crystal oscillator, a counter and a holding register.

RAID
RAID (redundant array of independent disks, originally redundant array of inexpensive disks) is a
storage technology that combines multiple disk drive components into a logical unit. Data is distributed
across the drives in one of several ways called "RAID levels", depending on what level of redundancy
and performance (via parallel communication) is required.

RAID Levels:
RAID 0 (block-level striping without parity or mirroring) has no (or zero) redundancy. It provides
improved performance and additional storage but no fault tolerance. Hence simple stripe sets are
normally referred to as RAID 0. Any drive failure destroys the array, and the likelihood of failure
increases with more drives in the array (at a minimum, catastrophic data loss is almost twice as likely
compared to single drives without RAID). A single drive failure destroys the entire array because when
data is written to a RAID 0 volume, the data is broken into fragments called blocks. The number of
blocks is dictated by the stripe size, which is a configuration parameter of the array. The blocks are
written to their respective drives simultaneously on the same sector. This allows smaller sections of the
entire chunk of data to be read off each drive in parallel, increasing bandwidth. RAID 0 does not
implement error checking, so any error is uncorrectable. More drives in the array means higher
bandwidth, but greater risk of data loss.
In **RAID 1** (mirroring without parity or striping), data is written identically to multiple drives, thereby producing a "mirrored set"; at least 2 drives are required to constitute such an array. While more constituent drives may be employed, many implementations deal with a maximum of only 2; of course, it might be possible to use such a limited level 1 RAID itself as a constituent of a level 1 RAID, effectively masking the limitation. The array continues to operate as long as at least one drive is functioning. With appropriate operating system support, there can be increased read performance, and only a minimal write performance reduction; implementing RAID 1 with a separate controller for each drive in order to perform simultaneous reads (and writes) is sometimes called *multiplexing* (or *duplexing* when there are only 2 drives).

In **RAID 2** (bit-level striping with dedicated Hamming-code parity), all disk spindle rotation is synchronized, and data is striped such that each sequential bit is on a different drive. Hamming-code parity is calculated across corresponding bits and stored on at least one parity drive.

In **RAID 3** (byte-level striping with dedicated parity), all disk spindle rotation is synchronized, and data is striped so each sequential byte is on a different drive. Parity is calculated across corresponding bytes and stored on a dedicated parity drive.

**RAID 4** (block-level striping with dedicated parity) is identical to RAID 5 (see below), but confines all parity data to a single drive. In this setup, files may be distributed between multiple drives. Each drive operates independently, allowing I/O requests to be performed in parallel. However, the use of a dedicated parity drive could create a performance bottleneck; because the parity data must be written to a single, dedicated parity drive for each block of non-parity data, the overall write performance may depend a great deal on the performance of this parity drive.

**RAID 5** (block-level striping with distributed parity) distributes parity along with the data and requires all drives but one to be present to operate; the array is not destroyed by a single drive failure. Upon drive failure, any subsequent reads can be calculated from the distributed parity such that the drive failure is masked from the end user. However, a single drive failure results in reduced performance of the entire array until the failed drive has been replaced and the associated data rebuilt. Additionally, there is the potentially disastrous RAID 5 write hole. RAID 5 requires at least 3 disks.

**RAID 6** (block-level striping with double distributed parity) provides fault tolerance of two drive failures; the array continues to operate with up to two failed drives. This makes larger RAID groups more practical, especially for high-availability systems. This becomes increasingly important as large-capacity drives lengthen the time needed to recover from the failure of a single drive. Single-parity RAID levels are as vulnerable to data loss as a RAID 0 array until the failed drive is replaced and its data rebuilt; the larger the drive, the longer the rebuild takes. Double parity gives additional time to rebuild the array without the data being at risk if a single additional drive fails before the rebuild is complete.

Source: [http://dayaramb.files.wordpress.com/2012/02/operating-system-pu.pdf](http://dayaramb.files.wordpress.com/2012/02/operating-system-pu.pdf)