OS Concepts & programming:-

==> OS Concepts & Design principles
==> Linux system programming(Linux internals)

==> Unix/Linux commands
==> Shell programming/scripting

Operating Systems:-
==> interface/glue/bridge/abstraction between hardware
        resources and user applications
==> Resource manager
==> Basis/platform for executing applications

can we design and execute apps in absence of OS,
direct access harware
==> platform dependency
==> ineeficient usage of resources
==> inconsistency or no healthy environment

Significance/advantages of OS:-
==> portable application coding
==> resource efficiency
==> healthy environment

Available OS in market:-

Windows, DOS
Linux, Unix
Mac OS
Chrome OS
Solaris
Plan9
OS/X

Android
iOS
Blackberry OS

Symbian
Tizen, BADA
Sailfish
Firefox OS
Firebase

Challenges in OS for portable devices:-

power consumption
thermal issues

constrained resources

OS Services:-

==> Process Management
==> Memory Management
==> File System Management -- logical
==> Storage/Disk Management -- physical
==> Network Management
==> I/O Device Management
==> Protection & Security

These services are provided by core of OS known as kernel to the applications in the form of system calls

----

GNU Linux

Unix -- initiated around 1970s
Minix -- Andrew S Tanenbaum
Linus Torvalds -- 1990s -- Linux kernel -- Public development

Around 1984 -- Richard M Stallman -- GNU -- free softwares
open source, freedom

GNU GPL -- GNU General Public License

other licenses:- Apache license, MIT license
Eclipse Public License
BSD license and many more

Creative Commons License -- non s/w components
-- documents, media etc.

Linux Distributions/Flavors:-
Debian -- Ubuntu,Mint,Kali, Backtrack
Redhat -- Fedora, CentOS
Suse, OpenSuse
Mandriva
and many more

Debian based -- deb packages, apt-get utility
Redhat based -- rpm packages...

Desktops:-
GNOME...unity ... gtk libs/C
KDE ... Qt libs/C++
MATE (based on old GNOME libs) ... gtk libs/C
LXDE, XFCE, LXQT etc... and many more

core for gtk or Qt libraries is X11 libraries

Apps:-

- Firefox,Chrome
- vlc
- GIMP, Inkscape
- Libreoffice
- Evince,okular..PDF
- gedit, kwrite/kate, pluma
- empathy,pidgin
- thunderbird

Developer:-

- vi, emacs, nano
- gcc/g++, gdb
- eclipse, netbeans,codeblocks
- mysql,postgresql,mongodb
- openjdk, tomcat
- mono libs
- virtualbox

Computer Architecture Basics:-

- CPU
- Memory
- Storage
- I/O Devices

System bus interconnects all these components

CPU:-

- Execution core -- ALU, CU
- Registers -- storage units on CPU chip itself
  limited in number & capacity
- Clock
- Cache -- small amount of memory on CPU chip itself
  -- frequenty,recently used code+data, needed
  in near future
- Levels of cache -- L1, L2 , L3
- Higher the cache level -- access speed goes down
  -- cost decreases
  -- capcity increases
- some levels are private to CPUs,some are shared among CPUs
  in SMP arch
- i-cache, d-cache

cache invalidation -- discard cache entries
-- rebuild the cache in future
cache contents are always additional copy to memory,
write back changes to memory during cache invalidation

Uniprocessor -- single CPU
SMP -- 2 or more CPUs on same chip(multi core)
or different chips
eg:- Core 2 Duo, Quad core,
Core i3,i5,i7
AMP -- heterogeneous cores..co processor
eg:- ARM+DSP(OMAP), GPUs, FPU
may be master--slave model
SMP+AMP

CPUs -- general purpose, special purpose(co processor)

registers:-

32 bit machine ==> register size
==> instruction set size (RISC)
==> data bus width -- word size
backward compatibility.....

Typical register:-
program counter(PC)/instruction pointer(IP)
        -- address of next instruction to be executed
program status word(PSW)/flags register
        -- bit of this register represents status of
           CPU exec or control CPU operations (status,control bits)
stack pointer, frame/base pointer -- track stack operations

general purpose registers -- for any purpose
accumulator has special significance even if it is of general purpose

x86 (intel 32 bit variants) regs:-
        EIP, EFLAGS, ESP, EBP
        EAX, EBX, ECX, EDX, ESI, EDI
x86_64(64 bit arch) regs:-
        RIP, RFLAGS,RAX etc.

mode bit in FLAGS register or PSW:- control bit

supervisor mode:-   privileged/unrestricted/unlimited
        entire access to hardware
        entire CPU instruction set
        entrire memory access
normal mode:-   restricted/limited/unprivileged
        zero or limited hardware acess
        subset of CPU instructions
        part of memory access

switching from normal mode to supervisor mode
can be acheived through a specialized CPU
instruction as "trap"
eg:- int 0x80, sysenter in intel
        swi,svc in arm
----
memory -- primary memory
             -- volatile
cost,speed,capacity lies b'n cpu,cache and disk

primary vs secondary memory?? CPU accessibility for load,store
                                                            operations

I/O Devices:-

```
        digital interface -- status registers
                          -- control registers
                          -- read registers
                          -- write registers
        I/O communication:-
                -- simple/direct i/o (impractical)
                -- interrupt driven i/o
                -- polling method using timers
```