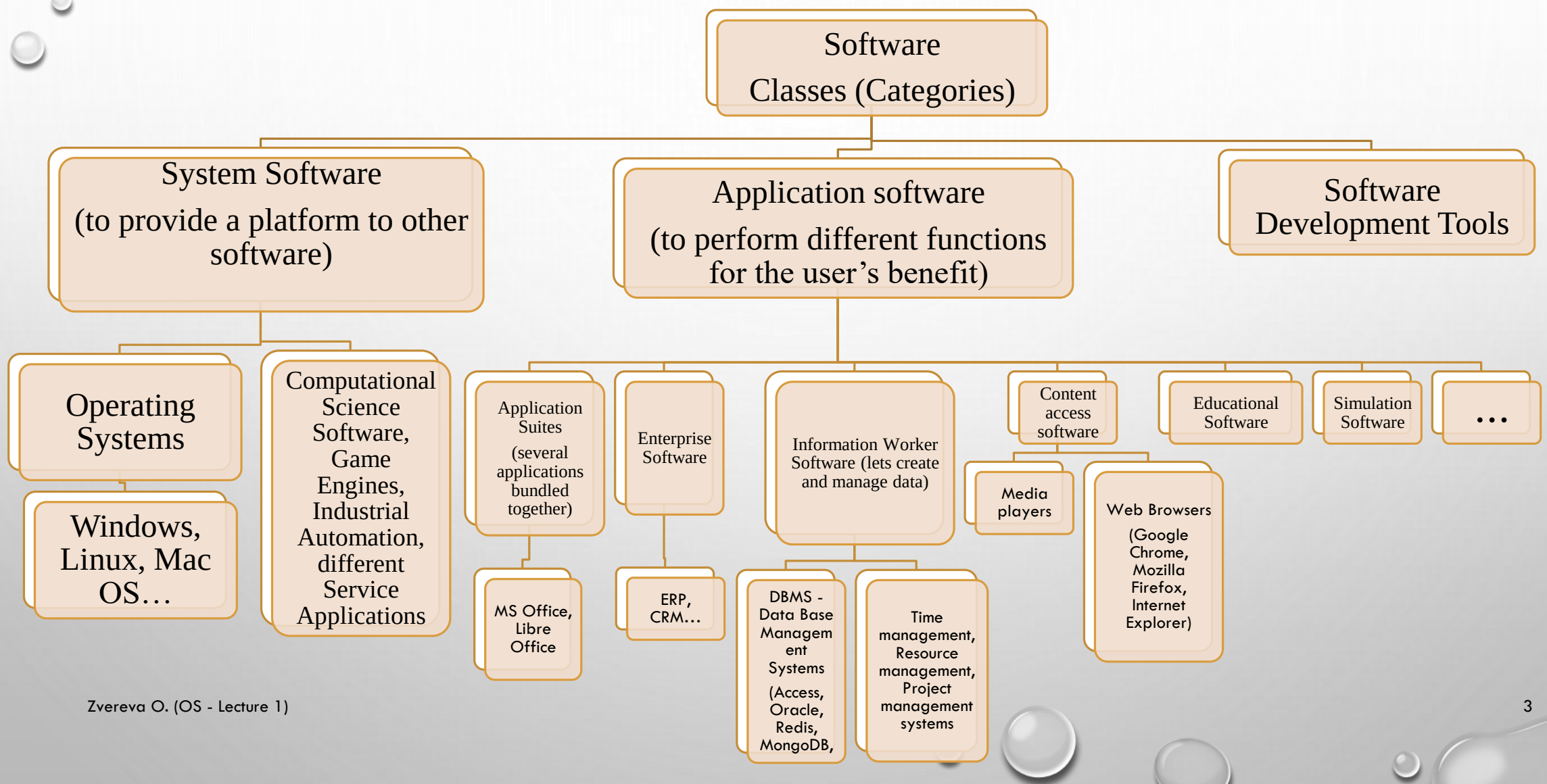
MODERN OPERATING SYSTEMS

LECTURER DR. OLGA MIKHAILOVNA ZVEREVA

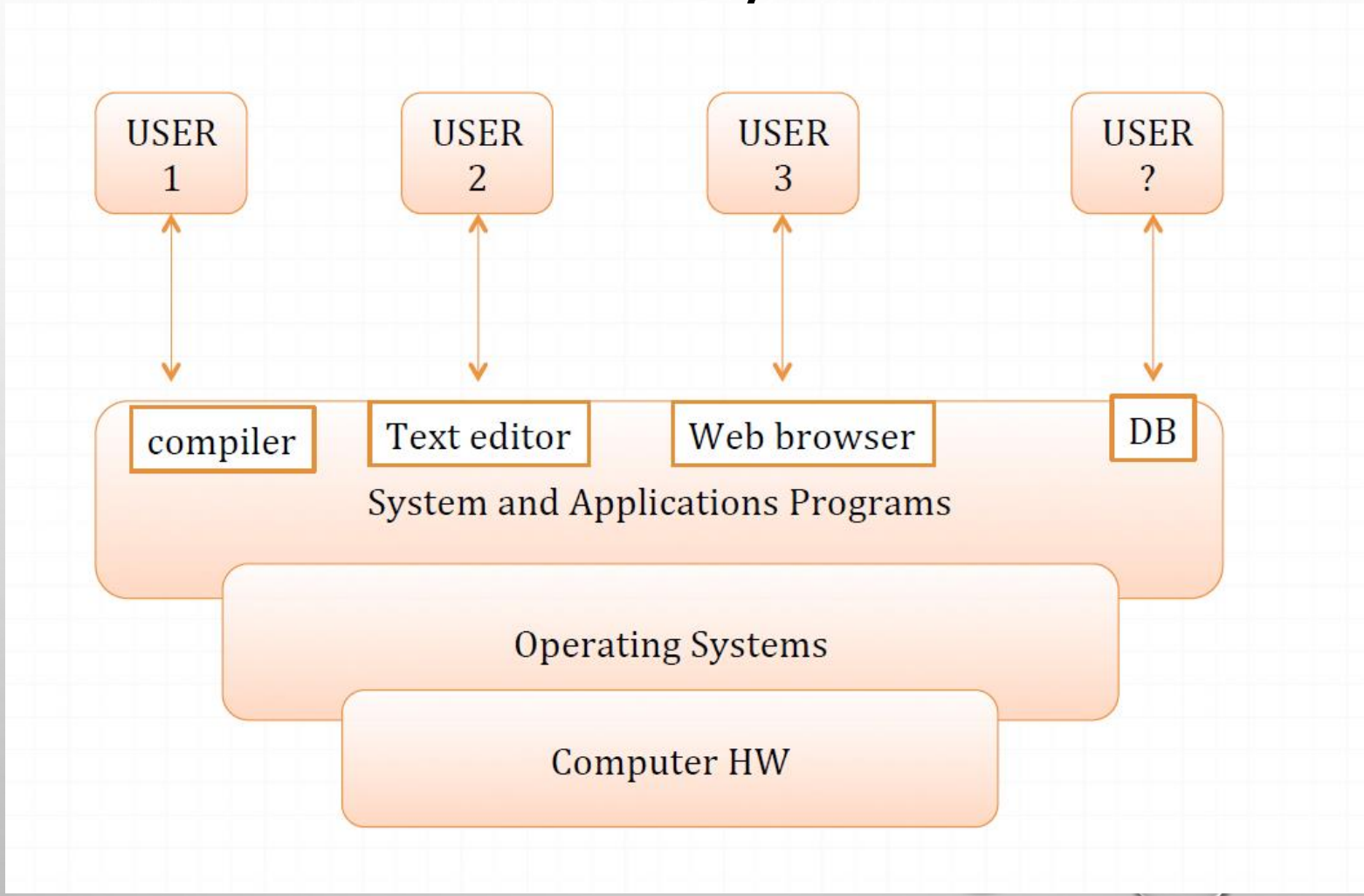
AGENDA

- OS DEFINITION
- HISTORY OF OS
- OS CLASSIFICATION
- FUNCTIONAL OS SUBSYSTEMS
  - PROCESS MANAGEMENT
  - MAIN MEMORY MANAGEMENT
  - I/O SYSTEM

SOFTWARE CLASSIFICATION



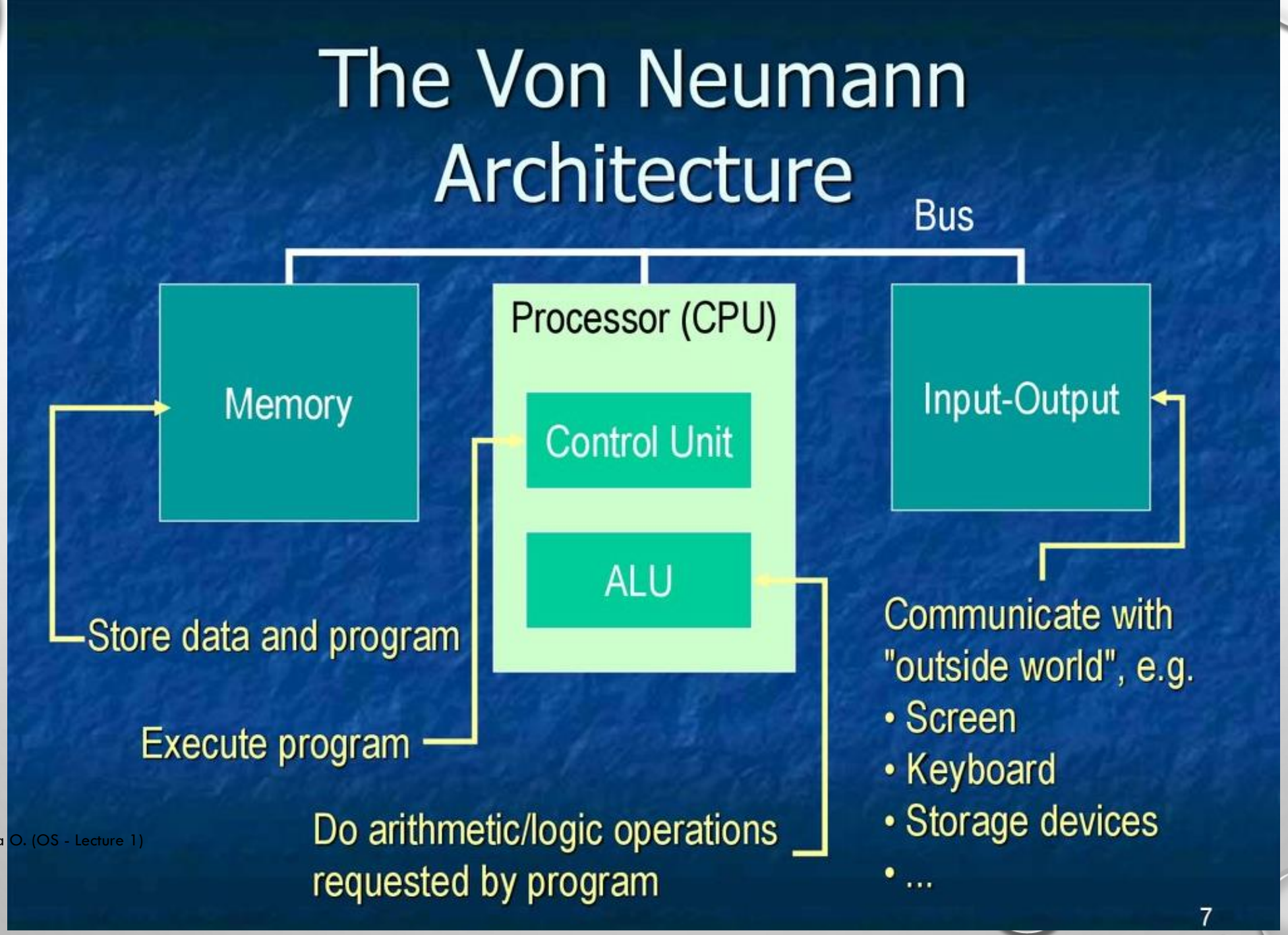
WHAT DOES AN OPERATING SYSTEM DO? (WORKS AS A MIDDLEWARE BETWEEN APPS AND HW)



OS DEFINITION

OS IS A SOFTWARE AND HARDWARE SYSTEM WHICH PERFORMS THE FOLLOWING TWO FUNCTIONS:

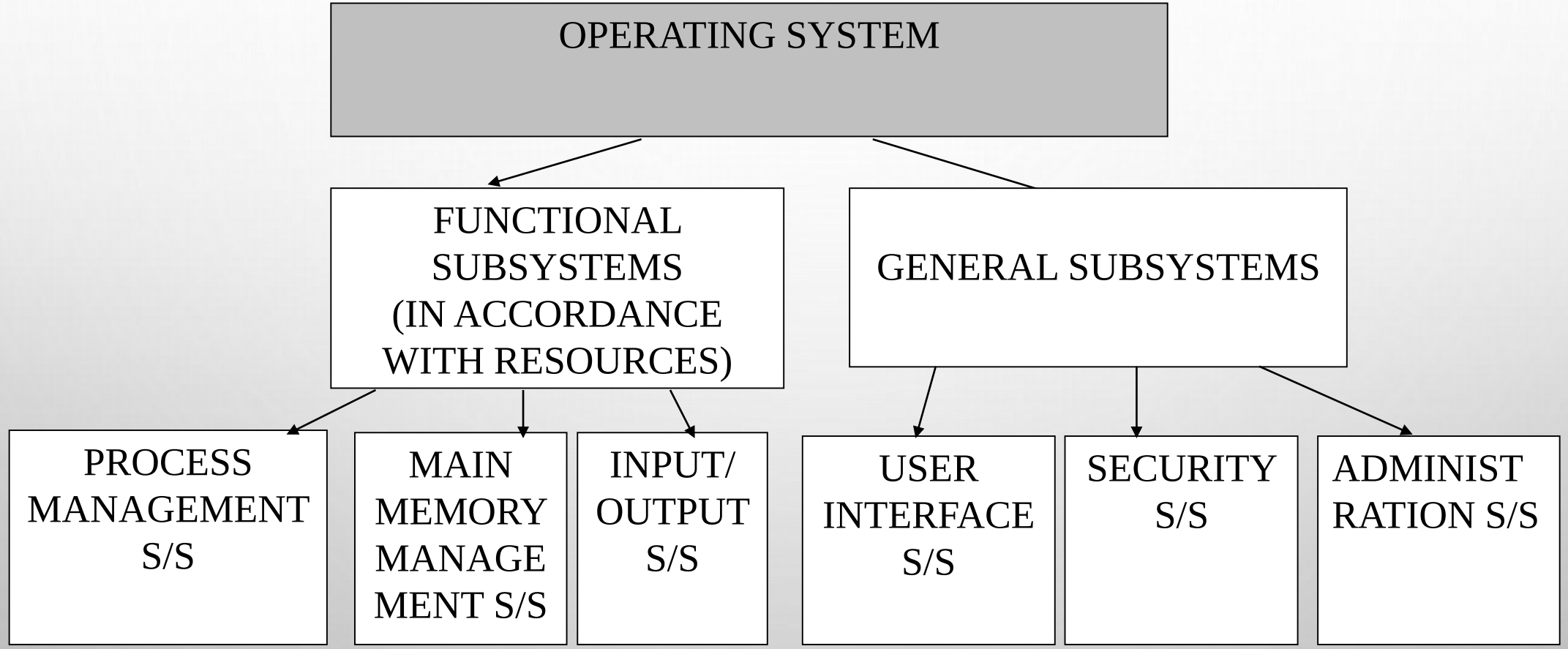
1. MANAGES ALL THE COMPUTER RESOURCES;
2. WORKS AS AN INTERFACE BETWEEN THE USER AND ITS APPLICATIONS FROM ONE SIDE AND HARDWARE COMPONENTS FROM THE OTHER.


The Von Neumann Architecture
Bus
Memory
Processor (CPU)
Control Unit
ALU
Input-Output
Store data and program
Execute program
Do arithmetic/logic operations requested by program
Communicate with "outside world", e.g.
• Screen
• Keyboard
• Storage devices
• ...

RESOURCES UNDER OS MANAGEMENT

- OS MANAGES MAIN MEMORY (RAM)
- OS MANAGES CPU
- OS MANAGES EXTERNAL DEVICES

OPERATING SYSTEM STRUCTURE (UPPER LOGICAL LEVEL)

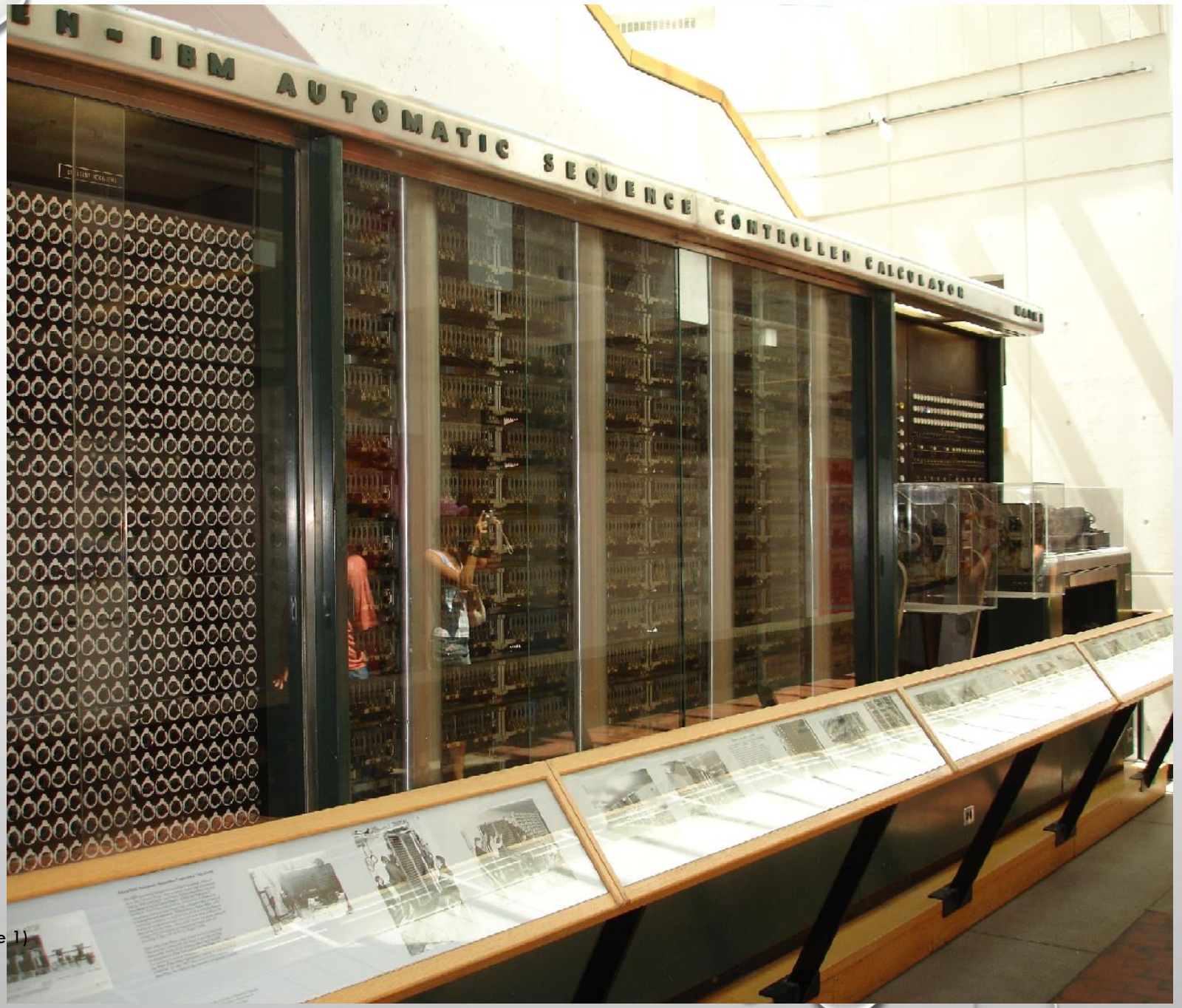


OS HISTORY

- HISTORY IS NOT ONLY AN INTERESTING STORY, BUT A USEFUL ONE AS WELL.
- INVESTIGATING TENDENCIES OF OS DEVELOPING IN THE PAST WE CAN PREDICT THEIR MAINSTREAMS IN THE FUTURE
- AS OS IS HIGHLY RELATED TO COMPUTER HARDWARE IT HAS BEEN EVOLVING IN ACCORDANCE WITH COMPUTER EVOLUTION.
- THE QUESTION IS: “HOW MANY GENERATIONS WERE ENCOUNTERED FOR HARDWARE? WHAT IS THE MAIN IDEA OF THIS DIVISION INTO GENERATIONS?”

1 ST GENERATION COMPUTERS (VACUUM TUBES AND PLUGBOARDS) 1945-1955

- THE EARLIEST ELECTRONIC DIGITAL COMPUTERS HAD NO OPERATING SYSTEMS.
- MACHINES OF THE TIME WERE SO PRIMITIVE THAT PROGRAMS WERE OFTEN ENTERED ONE BIT AT TIME ON ROWS OF MECHANICAL SWITCHES (PLUG BOARDS). PROGRAMMING LANGUAGES WERE UNKNOWN (NOT EVEN ASSEMBLY LANGUAGES).
- COMPUTERS COULDN'T DO ANY USEFUL WORK, WERE A KIND OF SCIENTIFIC PROJECT
- THE GENERAL MOTORS RESEARCH LABORATORIES IMPLEMENTED THE FIRST OPERATING SYSTEMS IN EARLY 1950'S FOR THEIR IBM 701.
- ACCORDING TO ANOTHER SOURCES, THE FIRST OS WAS DEVELOPED FOR MARK1 COMPUTER IN STANFORD UNIVERSITY.
- SYSTEMS GENERALLY RAN ONE JOB AT A TIME.


IBM AUTOMATIC SEQUENCE CONTROLLED CALCULATOR

2ND GENERATION (TRANSISTORS) (1955-1965)

- COMPUTERS BECAME **RELIABLE** AND THEY COULD BE MANUFACTURED TO DO SOME **REQUIRED** AND **USEFUL** WORK.
- NO ONE CAN AFFORD THOSE HIGH COST COMPUTERS EXCEPT **BIG CORPORATIONS** OR MAJOR **GOVERNMENT AGENCIES OR UNIVERSITIES.**
- THOSE SECOND GENERATION COMPUTERS WERE USED MOSTLY FOR THE SCIENTIFIC AND ENGINEERING CALCULATIONS. THEY WERE **PROGRAMMED IN FORTRAN** AND **ASSEMBLER.**
- FORTRAN WAS AN ALGORITHMIC (HIGH-LEVEL) LANGUAGE**, THE NECESSITY IN SYSTEM SOFTWARE** (COMPILERS, LINKERS, DIFFERENT LIBRARIES, AND ETC.) AROSE AND DIFFERENT SYSTEM PROGRAMS WERE CODED
- PROGRAM WAS REALIZED IN COMPUTER SYSTEM AS JOB AND JOB MANAGEMENT LANGUAGE WAS DEVELOPED TO CONTROL PROCESSES OF JOB FULFILLMENT
- OPERATING SYSTEMS: **FMS (**STANDS FOR FORTRAN MONITOR SYSTEM)**, IBSYS,** OPERATING SYSTEM OF IBM FOR THE 7094 (BATCH OS).

2ND GENERATION BATCH OS

TO RUN ANY JOB ON THOSE COMPUTERS:

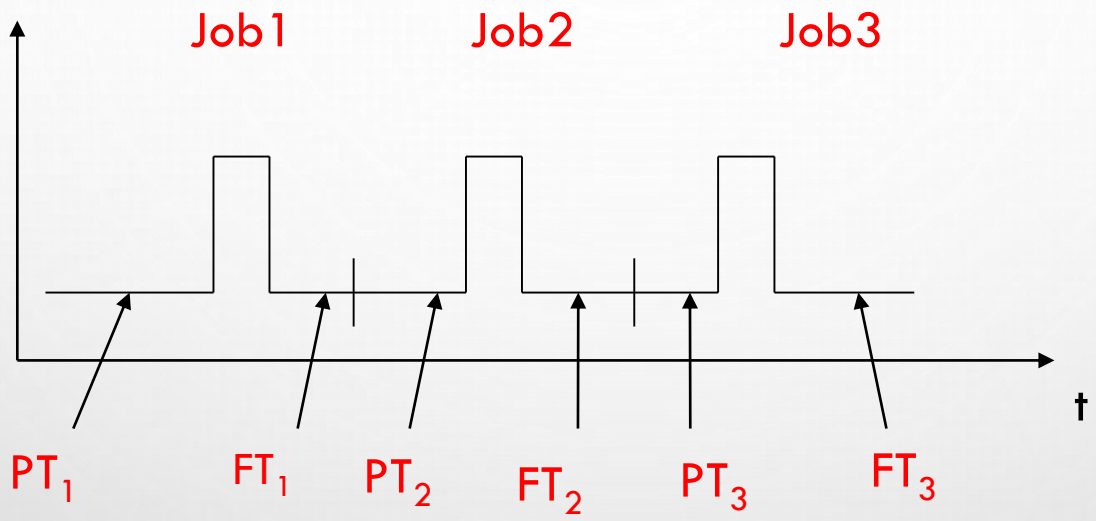
- A PROGRAMMER FIRST WRITES THE PROGRAM ON A PAPER USING FORTRAN OR ASSEMBLER,
- PUNCHES IT ON THE CARDS
- BRINGS THE CARD DECK DOWN TO THE INPUT ROOM AND HANDS IT TO ONE OF THE OPERATORS
- WHEN THE COMPUTER FINISHED WHATEVER JOB IT HAS BEEN CURRENTLY RUNNING, AN OPERATOR GOES OVER

TO THE PRINTER AND TEARS OFF THE OUTPUT OF THE PREVIOUS JOB

- THE OPERATOR TAKES THE CARD DECK THAT HAVE BEEN BROUGHT FROM THE INPUT ROOM AND READS IT IN (PUT INTO THE MAIN MEMORY)
- IF THE FORTRAN COMPILER IS NEEDED, THE OPERATOR GETS IT FROM A FILE CABINET AND READS IT IN.
- **CALCULATIONS START**
- ALL THE RESULTS ARE IN THE MAIN MEMORY
- OPERATOR HAS TO PRINT THE DUMP OF MAIN MEMORY AND BRING S IT TO THE PROGRAMMER
- MUCH COMPUTER TIME WAS WASTED (PRELIMINARY AND FINAL TIME)

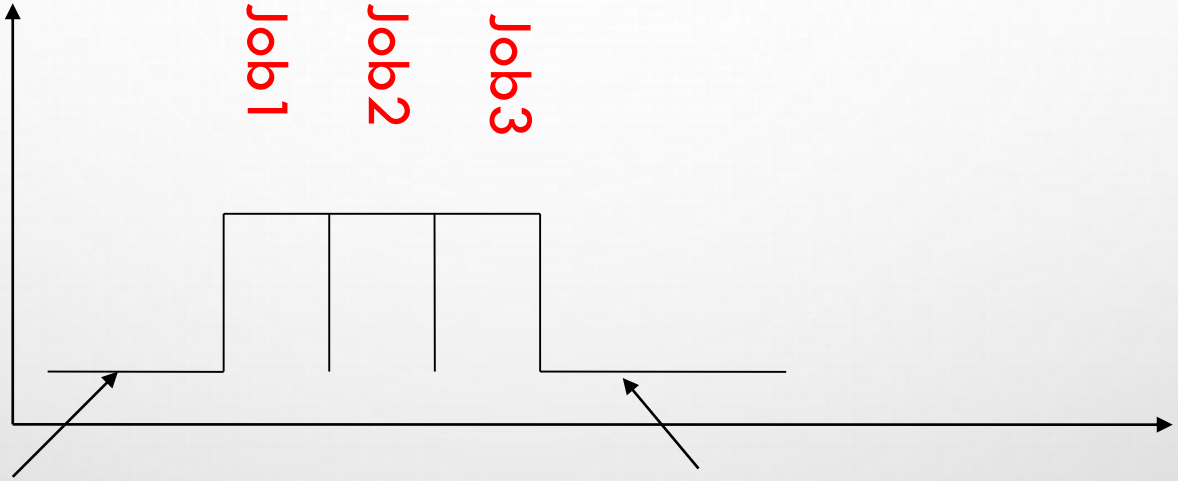
COMPUTER TIME WAS VERY EXPENSIVE , IT WAS NECESSARY TO CUT OFF ITS WASTES.

BEFORE BATCH OS



PT_i, FT_i – preliminary and final times of the i-th job

BATCH OS



Preliminary batch time (PBT) Final batch time (FBT)

$$PBT+FBT << \sum(PT_i + FT_i)$$

ADVANTAGES AND DISADVANTAGES OF BATCH OS

+ INCREASE EFFICIENCY OF COMPUTER TIME USAGE (CUT OFF TIME WASTE)

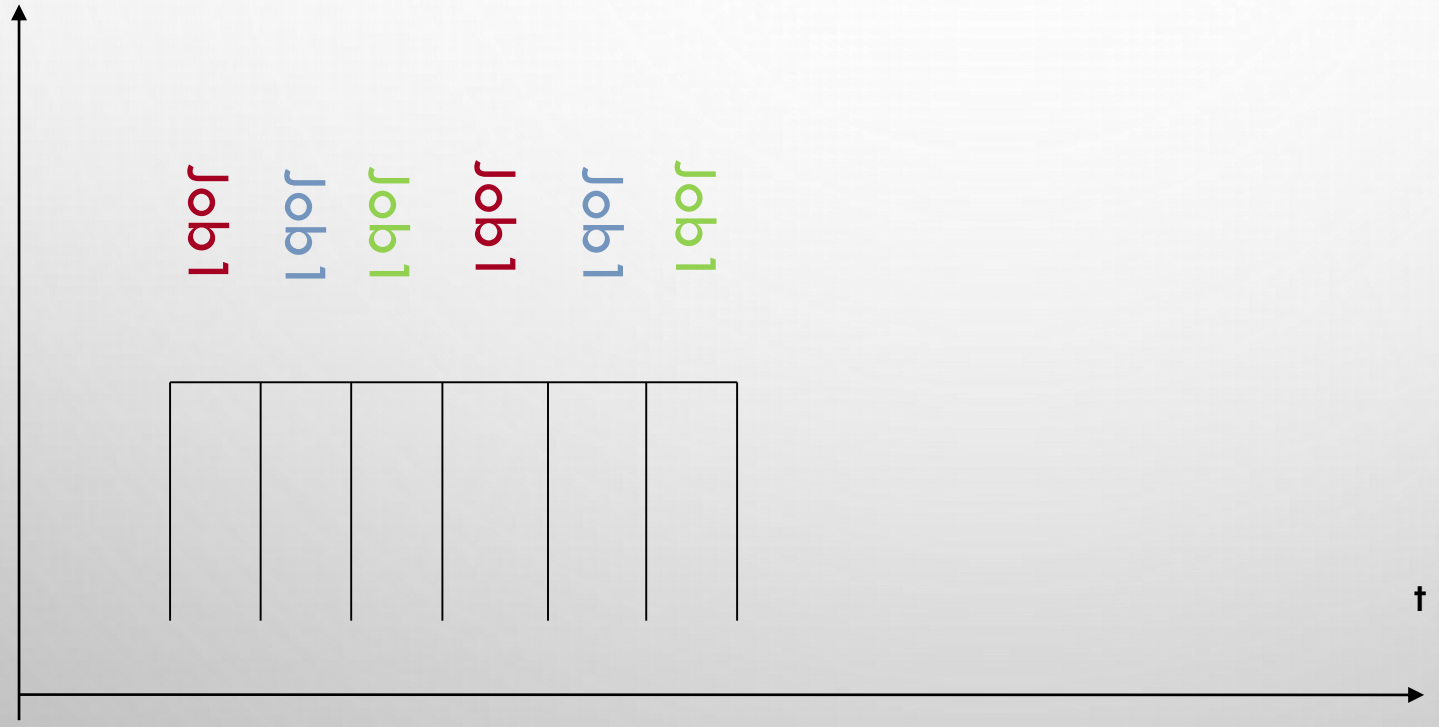
- POOR INTERFACE (USER CAN'T MANAGE CALCULATING PROCESS IN THE REAL TIME)

3RD GENERATION COMPUTERS (1965-1980)

THE RISE OF THE INTEGRATED CIRCUITS LAUNCHED THE POWER OF COMPUTERS, AND OPERATING SYSTEMS RESPONDED BY BECOMING INCREASINGLY COMPLEX AND OFFERING NEW TECHNIQUES.

- **MULTIPROGRAMMING:** IN THIS TECHNIQUE, THE MAIN MEMORY ALREADY HOLDS MORE THAN ONE PROGRAM, AND THE OPERATING SYSTEM IS RESPONSIBLE FOR ALLOCATING THE MACHINE'S RESOURCES TO EXECUTE TASKS BASED ON EXISTING NEEDS.
- **SPOOLING**: ABILITY TO READ JOBS FROM CARDS ONTO THE DISK AS SOON AS THEY WERE BROUGHT TO THE COMPUTER ROOM.
- **MULTIPROCESSING:** THE EXECUTION OF MULTIPLE CONCURRENT PROCESSING IN A SYSTEM, WITH EACH PROCESS RUNNING ON A SEPARATE CPU, AS OPPOSED TO A SINGLE PROCESS.
- **VIRTUAL MEMORY**: SYSTEM OF MODELING MAIN MEMORY WITH THE HELP DISK SPACE
- **FILE SYSTEM**
- DIFFERENT TYPE OS:
  - ✓BATCH
  - ✓TIME SHARING
  - ✓REAL TIME.

TIME SHARING SYSTEM



3RD GENERATION COMPUTERS (SOFTWARE-COMPATIBLE MACHINES, OR FAMILY OF COMPATIBLE COMPUTERS)

- IBM 360 FAMILY: COMPUTERS DIFFERED ONLY IN PRICE AND PERFORMANCE (MEMORY VOLUME, PROCESSOR SPEED, NUMBER OF I/O DEVICES PERMITTED, AND SO FORTH), RANGED FROM 1401-SIZED TO MUCH MORE POWERFUL 7094
- OPERATING SYSTEM - SYSTEM/360
- SINCE ALL THE MACHINES HAD THE SAME ARCHITECTURE AND INSTRUCTION SET, PROGRAMS WRITTEN FOR ONE MACHINE COULD RUN ON ALL THE OTHERS, AT LEAST IN THEORY
- THE 360 WAS DESIGNED TO HANDLE BOTH SCIENTIFIC (I.E., NUMERICAL) AND COMMERCIAL COMPUTING
- SUCCESSORS **-** THE 370, 4300, 3080, AND 3090 SERIES
- THE GREATEST STRENGTH OF THE “ONE FAMILY" IDEA WAS SIMULTANEOUSLY ITS GREATEST WEAKNESS. THE INTENTION WAS THAT ALL SOFTWARE, INCLUDING THE OPERATING SYSTEM, **OS/360** HAD TO WORK ON ALL MODELS (CONFLICTING REQUIREMENTS)
- THE RESULT WAS AN ENORMOUS AND EXTRAORDINARILY COMPLEX OPERATING SYSTEM, PROBABLY TWO TO THREE ORDERS OF MAGNITUDE LARGER THAN FMS. IT CONSISTED OF MILLIONS OF LINES OF ASSEMBLY LANGUAGE WRITTEN BY THOUSANDS OF PROGRAMMERS, AND CONTAINED THOUSANDS UPON THOUSANDS OF BUGS, WHICH NECESSITATED A CONTINUOUS STREAM OF NEW RELEASES IN AN ATTEMPT TO CORRECT THEM

3RD GENERATION COMPUTERS

- OTHER OPERATING SYSTEMS:
  - ✓ **CTSS** (COMPATIBLE TIME SHARING SYSTEM) - WAS DEVELOPED AT MIT (MASSACHUSETTS INSTITUTE OF TECHNOLOGY)
  - ✓ **MULTICS** (MULTIPLEXED INFORMATION AND COMPUTING SERVICE) – WAS PRODUCED BY MIT, BELL LABS, AND GENERAL ELECTRIC COMPANY, WAS DESIGNED TO SUPPORT HUNDREDS OF USERS ON A MACHINE ONLY SLIGHTLY MORE POWERFUL THAN AN INTEL 386-BASED PC (LACK OF COMMERCIAL SUCCESS)
- THE PHENOMENAL GROWTH OF MINICOMPUTERS, STARTING WITH THE DEC PDP-1
- **UNIX** OPERATING SYSTEM (KEN THOMPSON FROM BELL LABS FOR PDP-7 MINICOMPUTER, HAVING USED THE RESULTS OF MULTIX PROJECT)

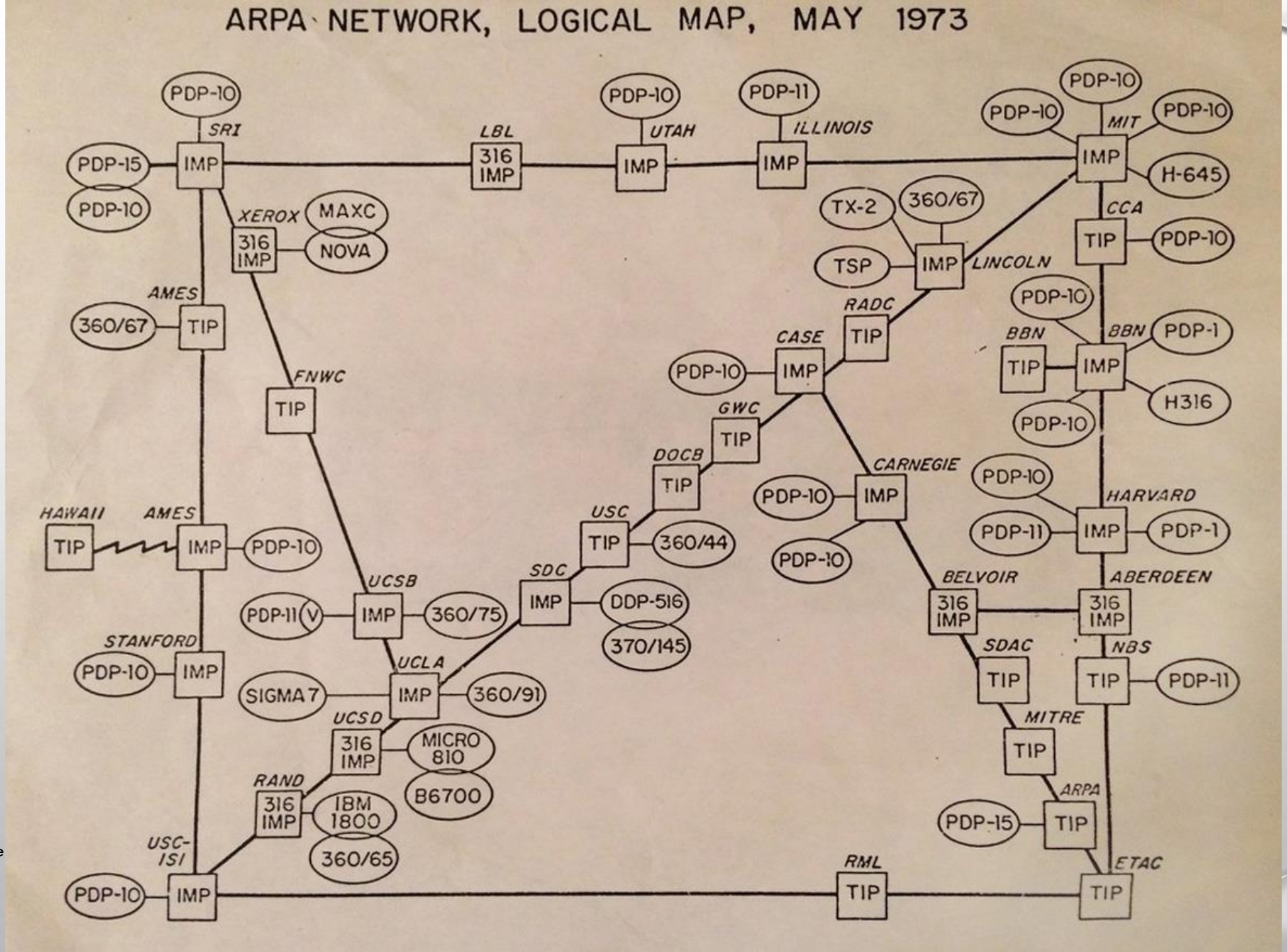
GLOBAL NETWORKS (ARPANET PROJECT)

- COMPUTERS BECAUSE OF THEIR COST ARE FEW BUT A LARGE CORPORATION COULD HAVE SEVERAL COMPUTERS AND GOVERNMENT (MILITARY) BODIES AS WELL.
- THE NECESSITY TO CONNECT THEM. GLOBAL NETWORKS WERE ENGINEERED BECAUSE THEY WERE GEOGRAPHICALLY DISTRIBUTED.
- ARPANET PROJECT STARTED THE ERA OF INTERNET
  - ✓ WAS INITIALLY FUNDED BY THE ADVANCED RESEARCH PROJECTS AGENCY (ARPA) OF THE UNITED STATES DEPARTMENT OF DEFENCE
  - ✓ BASED ON TWO MAIN TECHNICAL CONCEPTS: PACKET-SWITCHING THE TCP/IP PROTOCOL SUITE
  - ✓ ACCORDING TO STEPHEN J. LUKASIK, WHO AS DEPUTY DIRECTOR AND DIRECTOR OF DARPA THE GOAL WAS "TO EXPLOIT NEW COMPUTER TECHNOLOGIES TO MEET THE NEEDS OF MILITARY COMMAND AND CONTROL AGAINST NUCLEAR THREATS, ACHIEVE SURVIVABLE CONTROL OF US NUCLEAR FORCES, AND IMPROVE MILITARY TACTICAL AND MANAGEMENT DECISION MAKING".

✓ COLD WAR IN THE RELATIONS BETWEEN COUNTRIES RESULTED IN THE CONCEPT OF DECENTRALIZED CONTROL

(IF ONE OF THE NETWORK COMPUTERS HAD BEEN DESTROYED BECAUSE OF NUCLEAR ATTACK, NETWORK WOULD HAVE SURVIVED AND EXECUTED ALL THE NECESSARY OPERATIONS. ALTHOUGH TODAY THIS GOAL IS DEBATED AND SOMEONE CONSIDERED THIS PROJECT MORE RESEARCH THAN MILITARY.)

✓ LATER SEVERAL UNIVERSITIES' COMPUTERS WERE INCLUDED INTO THE NETWORK, THEN DIFFERENT COMPANIES, INDIVIDUAL USERS DID THE SAME


ARPA NETWORK, LOGICAL MAP, MAY 1973
PDP-10
SRI
PDP-15
IMP
PDP-10
LBL
316 IMP
PDP-10
UTAH
IMP
PDP-11
ILLINOIS
IMP
PDP-10
PDP-10
MIT
PDP-10
IMP
H-645
XEROX
316 IMP
MAXC
NOVA
TX-2
360/67
TSP
IMP
LINCOLN
CCA
TIP
PDP-10
AMES
360/67
TIP
RADC
TIP
PDP-10
BBN
TIP
BBN
IMP
PDP-1
H316
PDP-10
CASE
IMP
PDP-10
FNWC
TIP
GWC
TIP
DOCB
TIP
CARNEGIE
PDP-10
IMP
PDP-10
HAWAII
TIP
AMES
IMP
PDP-10
USC
TIP
360/44
SDC
IMP
DDP-516
370/145
PDP-10
PDP-10
HARVARD
PDP-11
IMP
PDP-1
UCSB
PDP-11
V
IMP
360/75
BELVOIR
316 IMP
ABERDEEN
316 IMP
STANFORD
PDP-10
IMP
UCLA
SIGMA 7
IMP
360/91
SDAC
TIP
NBS
TIP
PDP-11
UCSD
316 IMP
MICRO 810
B6700
MITRE
TIP
RAND
316 IMP
IBM 1800
360/65
ARPA
PDP-15
TIP
USC-ISI
PDP-10
IMP
RML
TIP
ETAC
TIP

4TH GENERATION COMPUTERS (1980-PRESENT)

- WITH THE DEVELOPMENT OF LSI (LARGE SCALE INTEGRATION) CIRCUITS, CHIPS CONTAINING THOUSANDS OF TRANSISTORS ON A SQUARE CENTIMETER OF SILICON, THE AGE OF THE PERSONAL COMPUTER DAWNED.
- IN 1974, WHEN INTEL CAME OUT WITH THE 8080, THE FIRST GENERAL-PURPOSE 8-BIT CPU, IT WANTED AN OPERATING SYSTEM (**CP/M - CONTROL PROGRAM FOR MICROCOMPUTERS**)
- 1981, AUGUST MIGHT BE CONSIDERED AS DATE OF BIRTH OF A PERSONAL COMPUTER.
- IBM DESIGNED THE IBM PC AND LOOKED AROUND FOR SOFTWARE TO RUN ON IT. PEOPLE FROM IBM CONTACTED BILL GATES.
- GATES REALIZED THAT A LOCAL COMPUTER MANUFACTURER, SEATTLE COMPUTER PRODUCTS, HAD A SUITABLE OPERATING SYSTEM, **DOS** (**DISK OPERATING SYSTEM**). HE APPROACHED THEM AND ASKED TO BUY IT (FOR $50,000), WHICH THEY READILY ACCEPTED.
- WITH TIM PATTERSON THEY REVISED THIS SYSTEM AND RENAMED IT AS **MS-DOS** (**MICROSOFT DISK OPERATING SYSTEM**)
- FIRST, MS-DOS AND UNIIX OCCUPIED THE WHOLE MARKET OF OS

MS-DOS

- HAD SEVERAL VERSIONS IN THE FORMAT: XX.XX (I.E. 6.1, 6.22)
- SINGLE PROGRAM
- SINGLE USER
- COMMAND LINE INTERFACE
- WAS ABLE TO START FROM DISKETTE (1.2 MB)
- HAD UNIX LIKE FILE SYSTEM
- WITHOUT NETWORKING FUNCTIONS

GUI

- DOUG ENGELBART INVENTED GRAPHICAL USER INTERFACE (GUI) AT STANFORD RESEARCH INSTITUTE IN THE 1960-S COMPLETE WITH WINDOWS, ICONS, MENUS, AND MOUSE.
- THESE IDEAS WERE ADOPTED BY RESEARCHERS AT XEROX PARC AND INCORPORATED INTO MACHINES THEY BUILT.
- STEVE JOBS, WHO CO-INVENTED THE APPLE COMPUTER IN HIS GARAGE, VISITED PARC, SAW A GUI, AND INSTANTLY REALIZED ITS POTENTIAL VALUE
- JOBS THEN EMBARKED ON BUILDING AN APPLE WITH A GUI.
- THIS PROJECT LED TO THE LISA (COMPUTER NAME), WHICH WAS TOO EXPENSIVE AND FAILED COMMERCIALLY.
- JOBS' SECOND ATTEMPT (THE APPLE MACINTOSH) WAS A HUGE SUCCESS: IT WAS MUCH CHEAPER THAN THE LISA AND IT WAS **USER FRIENDLY** (IT WAS INTENDED FOR USERS WHO NOT ONLY KNEW NOTHING ABOUT COMPUTERS BUT HAD ABSOLUTELY NO INTENTION TO LEARN).

WINDOWS

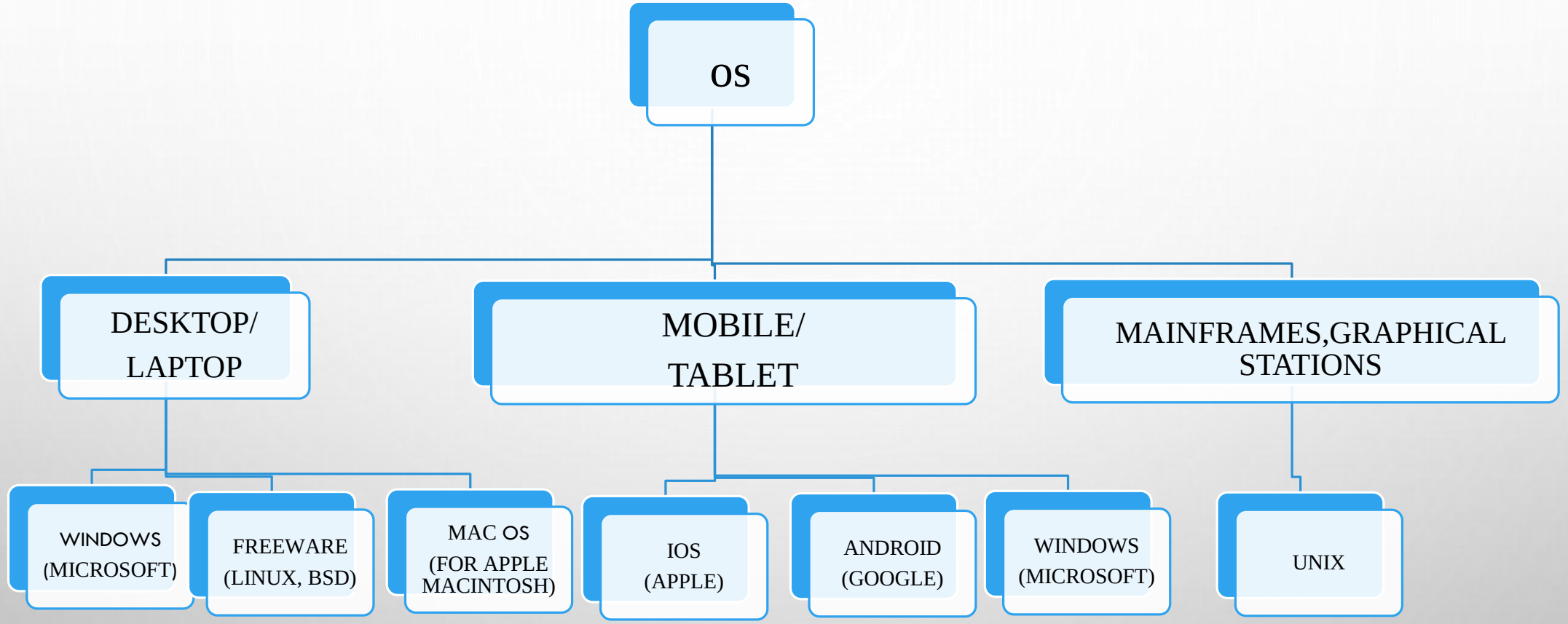
- STRONGLY INFLUENCED BY THE SUCCESS OF THE MACINTOSH
- GUI-BASED SYSTEM
- VERSIONS:
  - WINDOWS 3.XX (3.10, 3.11):
    - ✓ RAN ON TOP OF MS-DOS
    - ✓ JUST A GRAPHICAL ENVIRONMENT ON TOP OF MS-DOS
  - WINDOWS 95/98
    - ✓ REAL OS, INCORPORATED MANY OPERATING SYSTEM FEATURES INTO IT, USING THE UNDERLYING MS-DOS SYSTEM ONLY FOR BOOTING AND RUNNING OLD MS-DOS PROGRAMS
    - ✓ A LARGE AMOUNT OF 16-BIT INTEL ASSEMBLY LANGUAGE
  - WINDOWS NT (NT STANDS FOR NEW TECHNOLOGY):
    - ✓ IS COMPATIBLE WITH WINDOWS 95 AT A CERTAIN LEVEL, BUT A COMPLETE REWRITE FROM SCRATCH INTERNALLY
    - ✓ FIRST VERSIONS:
      - ❖ FOR WORKSTATIONS : WINDOWS NT 4.0 WORKSTATION, WINDOWS 2000 PROFESSIONAL, WINDOWS XP PROFESSIONAL;
      - ❖ FOR HOME COMPUTERS: WINDOWS XP HOME EDITION
      - ❖ FOR SERVERS: WINDOWS NT 4.0 SERVER, WINDOWS 2000 SERVER(ADVANCED SERVER, DATACENTER SERVER), WINDOWS 2003 SERVER
  - WINDOWS VISTA

WINDOWS

- WINDOWS 7

- WINDOWS 8
- WINDOWS 10

OS CLASSIFICATION



Operating System Share by Version

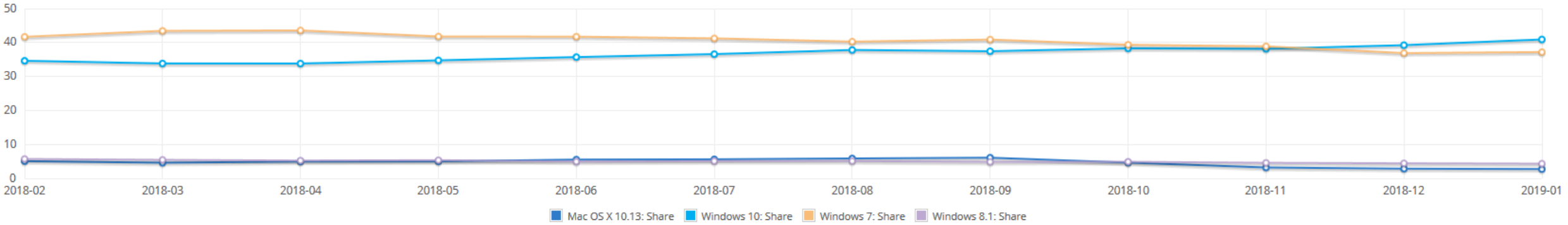
Monthly | 2018-02 | to | 2019-01 | Run

AND | OR

+ Add filter | + Add group

Device Type | in | Desktop/laptop ×

✖ Delete



Show 10 entries

Search:

Platform Version	Share
Windows 7	40.41%
Windows 10	36.94%
Windows 8.1	4.97%
Mac OS X 10.13	4.58%
Windows XP	3.96%
Mac OS X 10.12	1.59%
Mac OS X 10.14	1.51%
Linux	1.49%

Operating System Share by Version

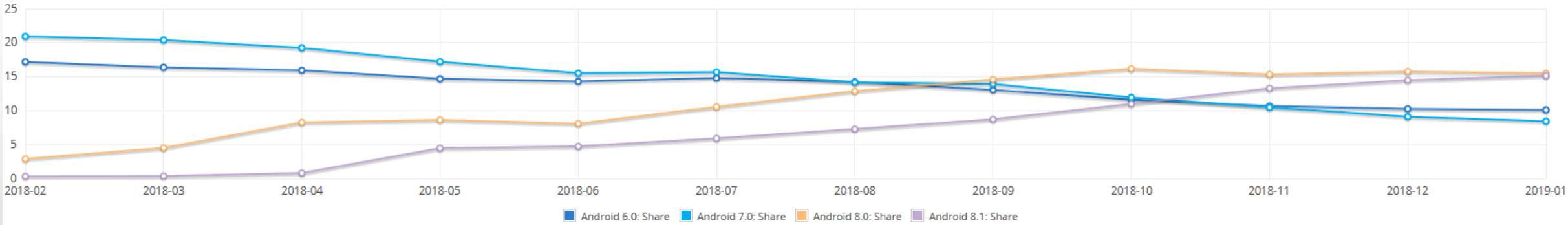
Monthly | 2018-02 | to | 2019-01 | Run

AND | OR

Add filter | Add group

Device Type | in | Mobile ×

Delete



Show 10 entries

Search:

Platform Version	Share
Android 7.0	14.04%
Android 6.0	13.19%
Android 8.0	11.89%
Android 8.1	8.15%
Android 7.1	6.92%
Android 5.1	6.88%
iOS 11.4	6.71%
iOS 12.1	5.16%
iOS 11.2	4.87%
iOS 11.3	3.76%

Operating System Share by Version

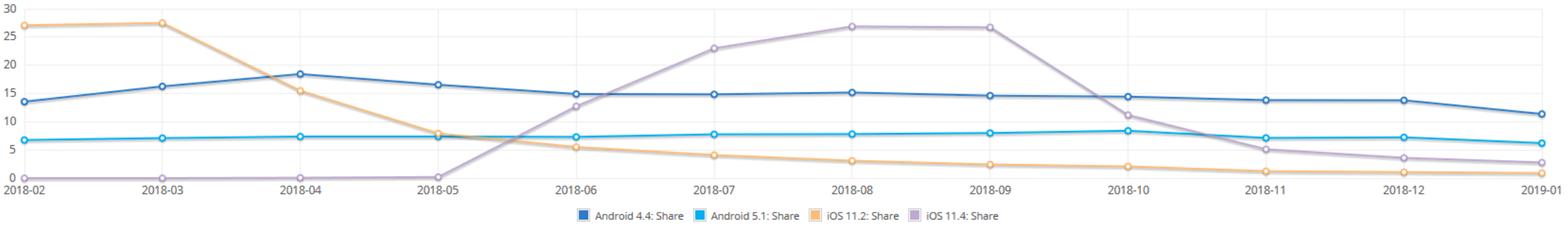
Monthly | 2018-02 | to | 2019-01 | Run

AND | OR

Add filter | Add group

Device Type | in | Tablet ×

Delete



Show 10 entries

Search:

Platform Version	Share
Android 4.4	14.72%
iOS 11.4	8.13%
iOS 11.2	7.64%
Android 5.1	7.31%
Android 7.0	6.89%
iOS 9.3	6.72%
Android 6.0	5.75%
iOS 10.3	5.21%
iOS 12.1	5.12%
iOS 11.3	4.84%

OS CLASSIFICATION

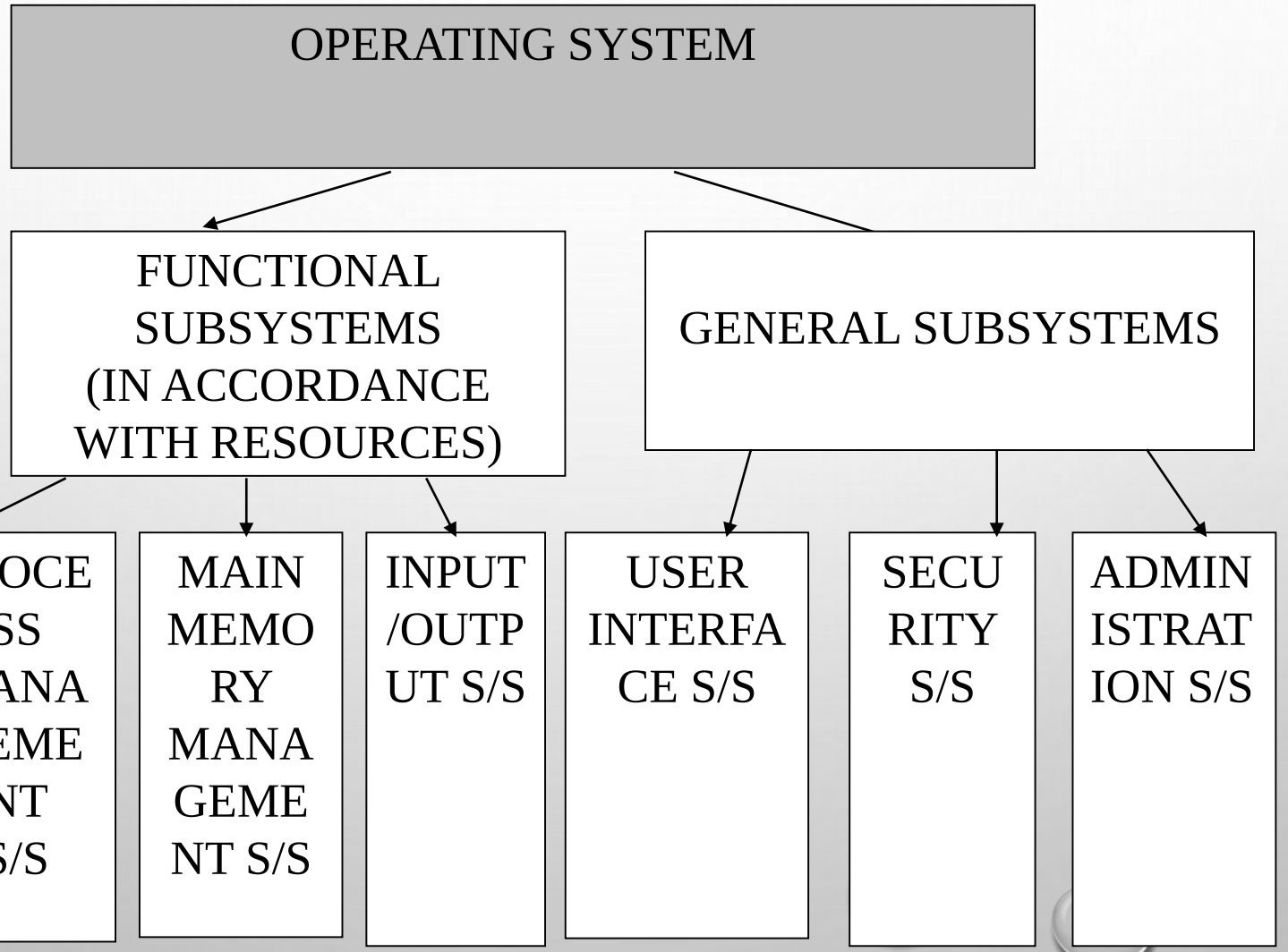
PROPRIETARY

- USERS CAN'T STUDY, CHANGE AND SHARE CODE
- PRODUCED BY A COMPANY (MICROSOFT, GOOGLE, APPLE, ETC.)
- MUST BUY LICENSES

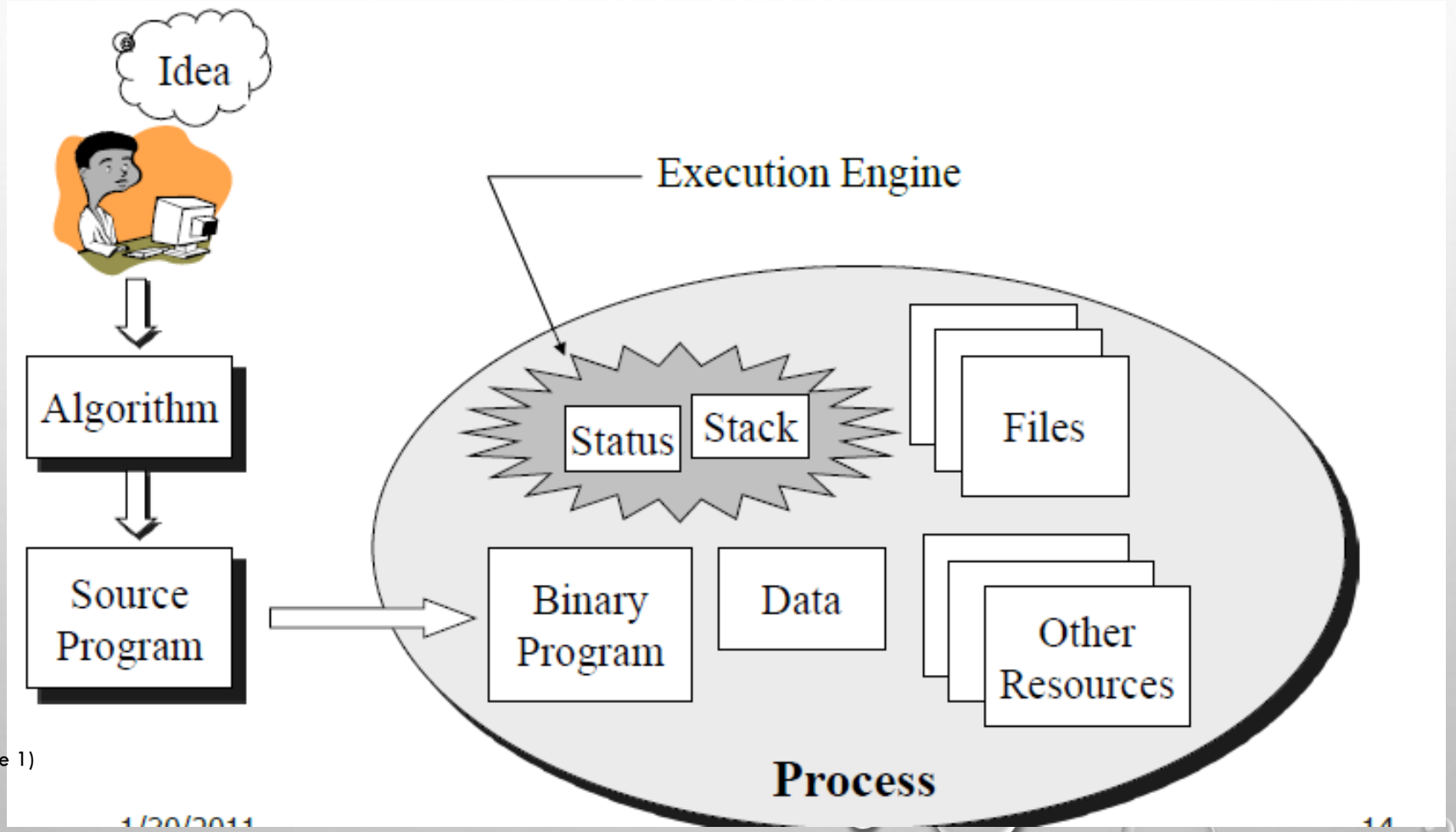
FREE

- FREEDOM 0: RUN THE PROGRAM FOR ANY PURPOSE
- FREEDOM 1: STUDY AND CHANGE THE PROGRAM
- FREEDOM 2: REDISTRIBUTE AND MAKE COPIES
- FREEDOM 3: IMPROVE THE PROGRAM AND RELEASE IMPROVEMENTS

OPERATING SYSTEM STRUCTURE (UPPER LOGICAL LEVEL) COPY OF SLIDE 7



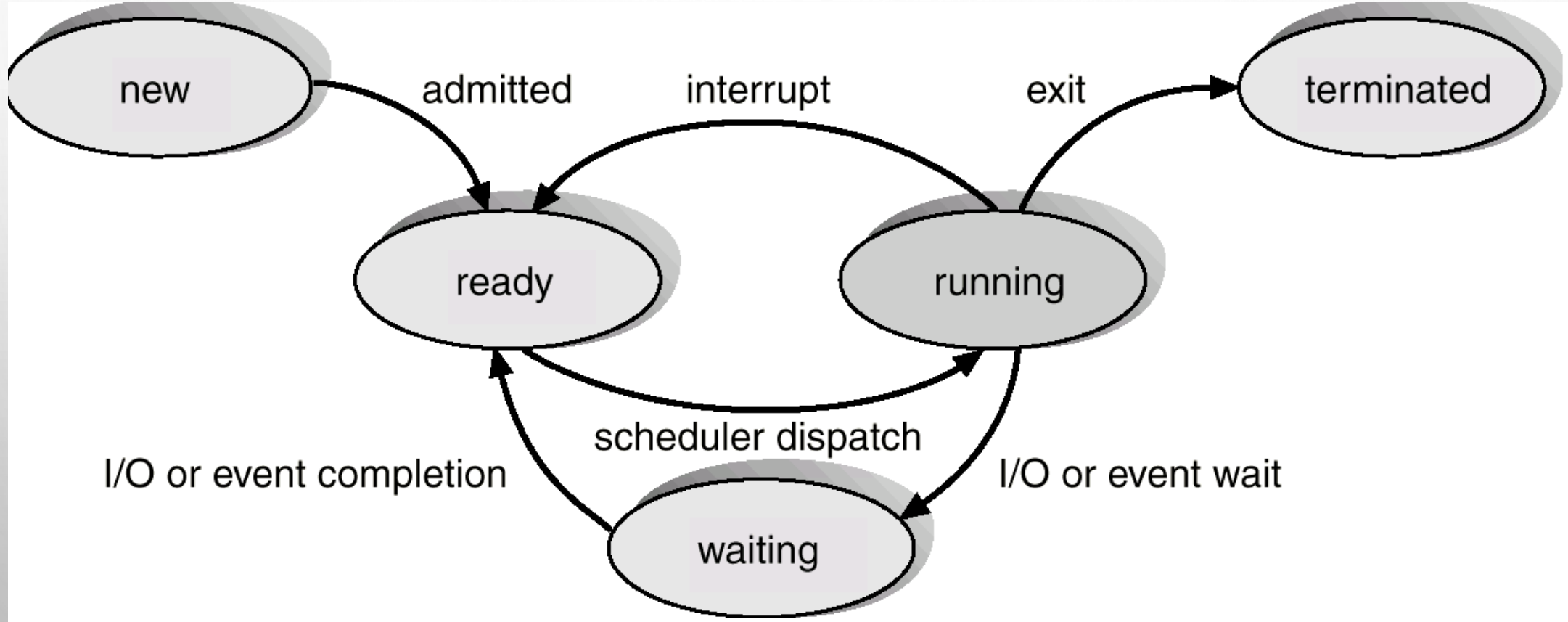
PROCESS MANAGEMENT: ALGORITHM, PROGRAM, PROCESS



OS MANAGES CPU

- OS PROVIDES THE PROCESS ABSTRACTION
  - PROCESS IS A RUNNING PROGRAM (PROGRAM IS STATIC, PROCESS IS DYNAMIC)
  - PROCESS IS A REQUEST FOR RESOURCES
- OS CREATES AND MANAGES PROCESSES
- MODERN SYSTEMS CONSIST OF A COLLECTION OF PROCESSES (MULTIPROGRAMMING)
  - ✓ EACH PROCESS HAS THE ILLUSION OF HAVING THE COMPLETE CPU
  - ✓ OS VIRTUALIZES CPU
  - ✓ TIME SHARES CPU BETWEEN PROCESSES
  - ✓ ENABLES COORDINATION BETWEEN PROCESSES

PROCESS STATE DIAGRAM

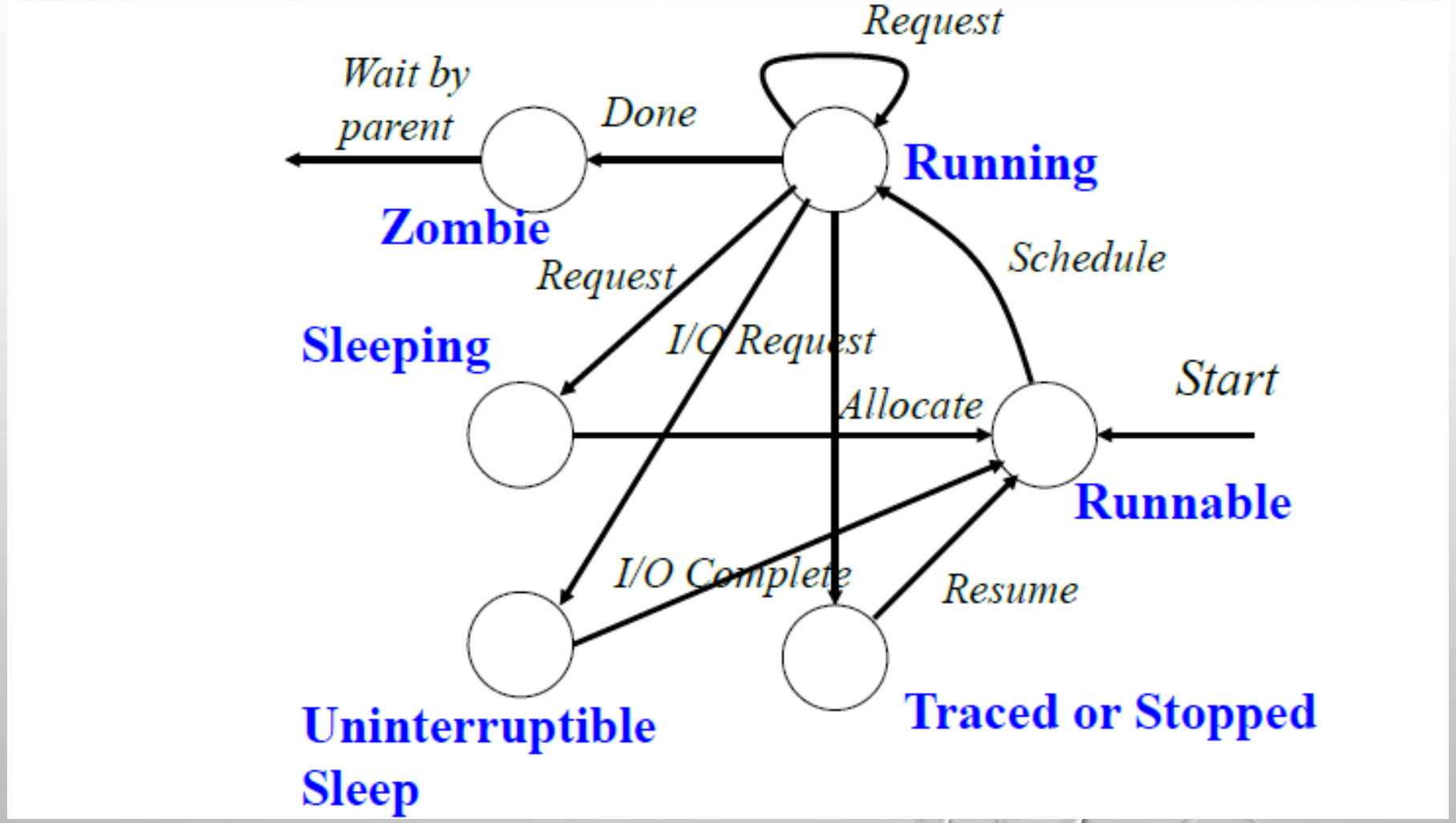


PROCESS STATES

AS A PROCESS EXECUTES, IT CHANGES STATE:

- **NEW**: THE PROCESS IS BEING CREATED
- **READY**: THE PROCESS IS WAITING TO BE ASSIGNED TO A PROCESSOR
- **RUNNING**: INSTRUCTIONS ARE BEING EXECUTED
- **WAITING**: THE PROCESS IS WAITING FOR SOME EVENT TO OCCUR
- **TERMINATED**: THE PROCESS HAS FINISHED EXECUTION

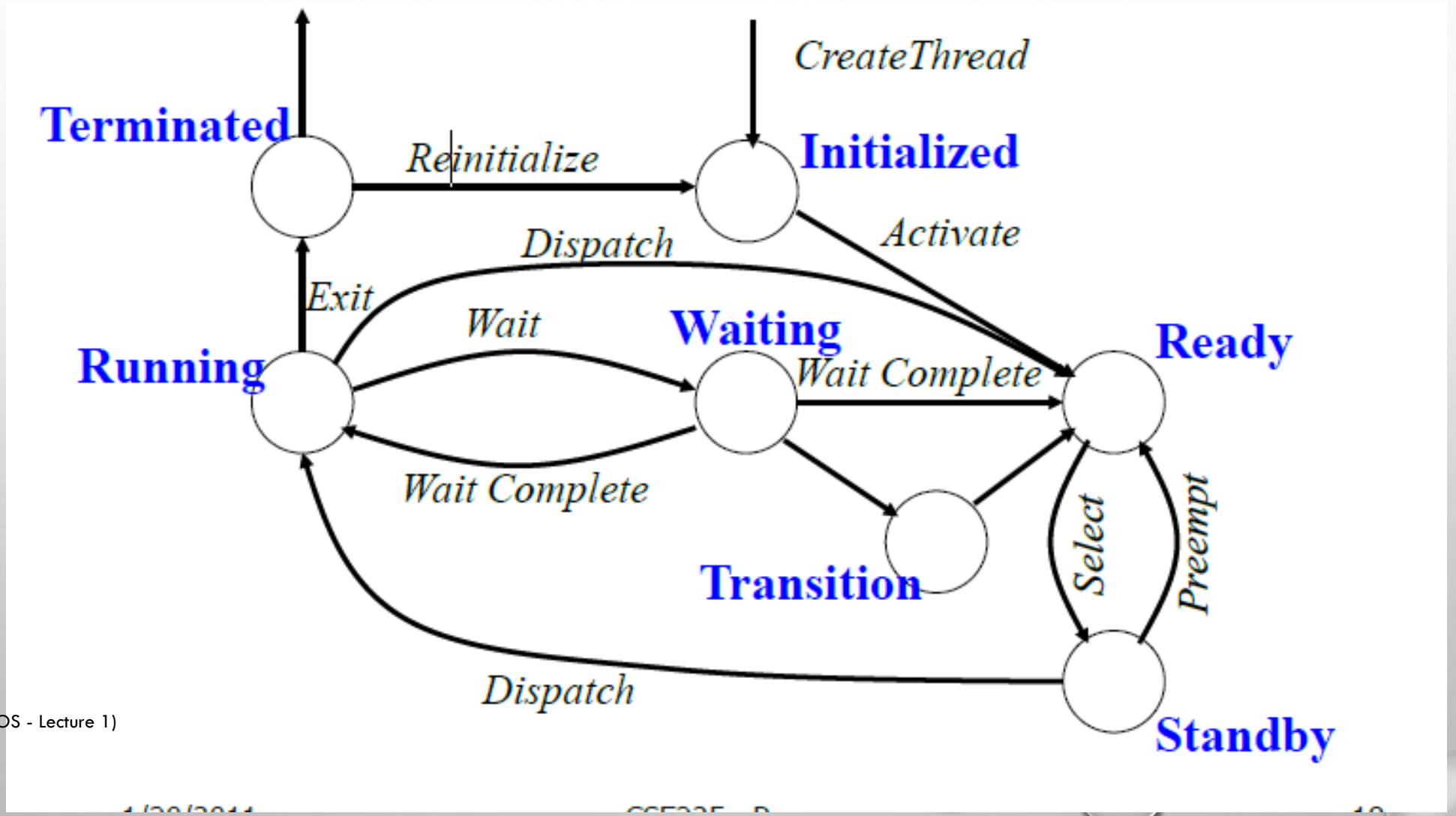
PROCESS STATE DIAGRAM IN UNIX (LINUX)



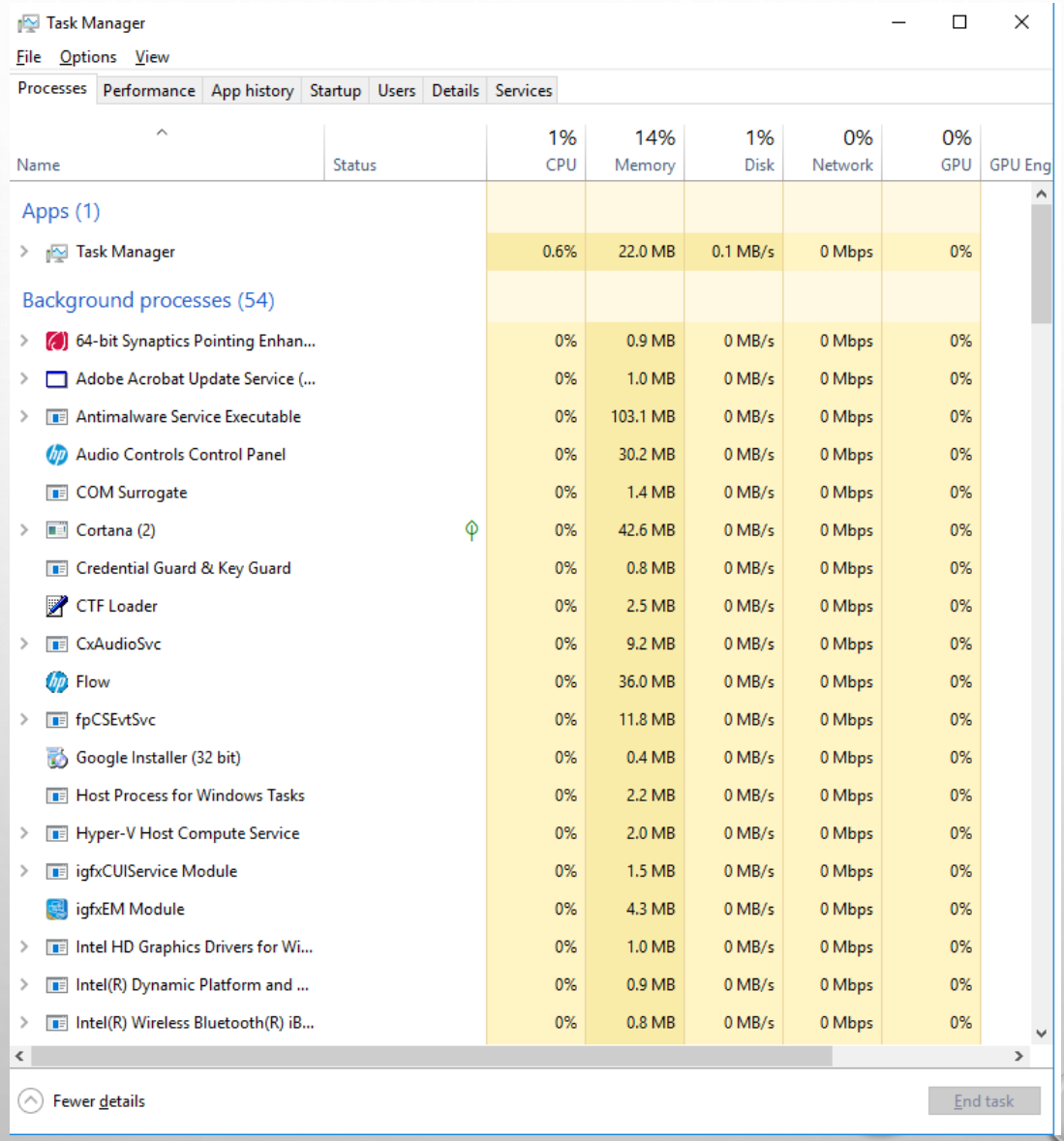
PROCESS LIST IN LINUX

```
[user@localhost user]$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4   1580   468 ?        S    14:33   0:02 init [5]
root         2  0.0  0.0      0     0 ?        SWN  14:33   0:00 [ksoftirqd/0]
root         3  0.0  0.0      0     0 ?        SW<  14:33   0:00 [events/0]
root         4  0.0  0.0      0     0 ?        SW<  14:33   0:00 [kblockd/0]
root         5  0.0  0.0      0     0 ?        SW   14:33   0:00 [kapmd]
root         6  0.0  0.0      0     0 ?        SW   14:33   0:00 [pdflush]
root         7  0.0  0.0      0     0 ?        SW   14:33   0:00 [pdflush]
root         8  0.0  0.0      0     0 ?        SW   14:33   0:01 [kswapd0]
root         9  0.0  0.0      0     0 ?        SW<  14:33   0:00 [aio/0]
root        11  0.0  0.0      0     0 ?        SW   14:33   0:00 [kseriod]
root        15  0.0  0.0      0     0 ?        SW   14:33   0:00 [scsi_eh_0]
root        19  0.0  0.0      0     0 ?        SW   14:33   0:01 [kjournald]
root       115  0.0  1.1   2048  1116 ?        S    14:33   0:00 devfsd /dev
root       246  0.0  0.0      0     0 ?        SW   14:33   0:00 [khubd]
root       417  0.0  0.0      0     0 ?        SW   14:33   0:00 [kjournald]
root       832  0.0  0.5   1652   548 ?        S    14:33   0:00 /sbin/ifplugd -w
rpc        858  0.0  0.6   1712   580 ?        S    14:33   0:00 portmap
root       882  0.0  0.6   1640   584 ?        S    14:33   0:00 syslogd -m 0
root       890  0.0  1.3   2560  1304 ?        S    14:33   0:00 klogd -2
xfs       1086  0.0  1.9   5160  1820 ?        S    14:33   0:00 xfs -port -1 -dae
root      1158  0.0  0.7   2688   676 ?        S    14:33   0:00 /usr/bin/mdkkdm -
daemon    1184  0.0  0.5   1616   536 ?        S    14:33   0:00 /usr/sbin/atd
```

THREAD STATE DIAGRAM IN WINDOWS NT



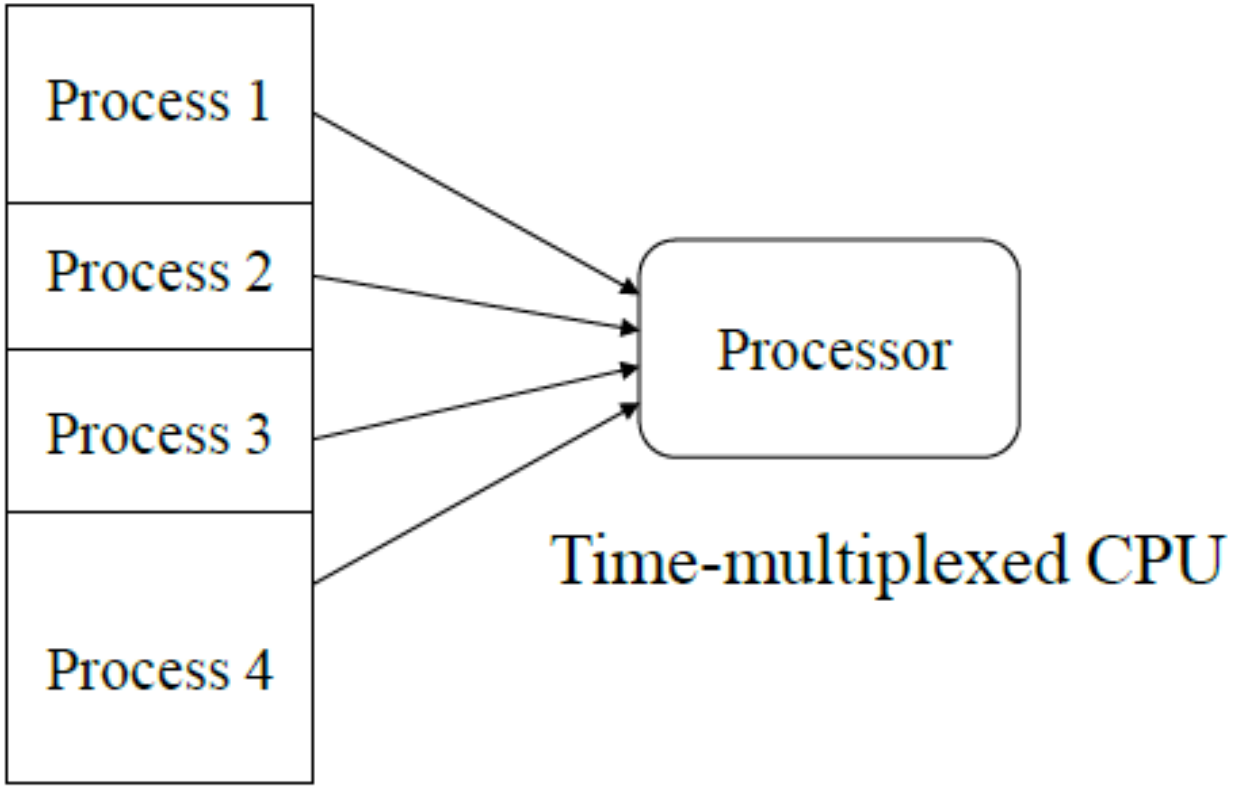
WINDOWS TASK MANAGER



THREAD CONCEPT

- THE PROCESS CONCEPT INCORPORATES TWO ABSTRACTIONS:
  - ✓ A VIRTUAL PROCESSOR (AN EXECUTION CONTEXT)
  - ✓ A RESOURCE OWNERSHIP (AN ADDRESS SPACE, OPEN FILES ETC..)
- WITHIN AN ADDRESS SPACE, WE CAN HAVE MORE UNITS OF EXECUTION: THREADS
- ALL THE THREADS OF A PROCESS SHARE THE SAME ADDRESS SPACE AND THE SAME RESOURCES

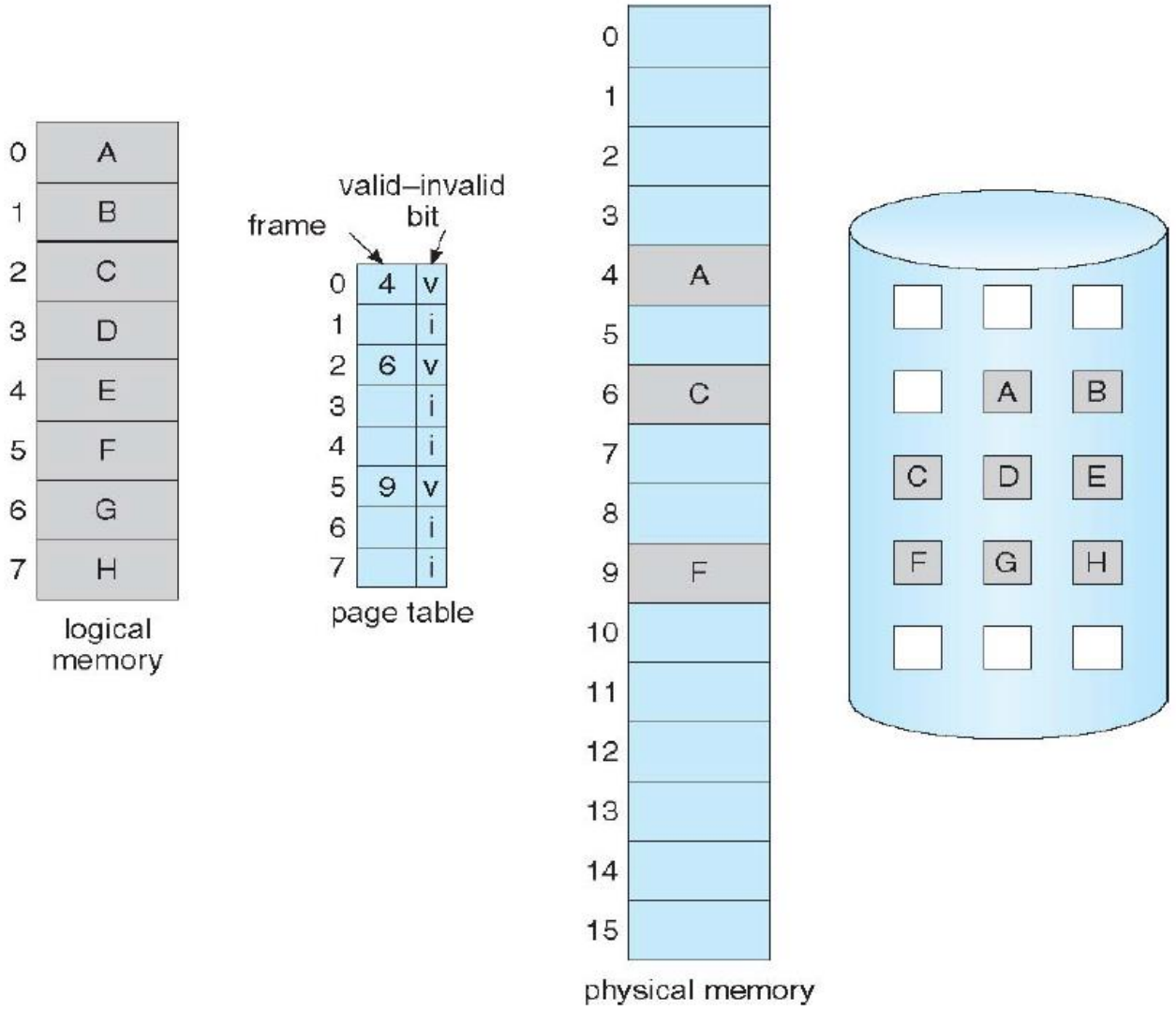
HOW MULTIPROGRAMMING WORKS



MAIN MEMORY

- PROGRAM MUST BE BROUGHT (FROM DISK) INTO MEMORY AND PLACED WITHIN A PROCESS FOR IT TO BE RUN. MAIN MEMORY AND REGISTERS ARE ONLY STORAGE CPU CAN ACCESS DIRECTLY
- THE OPERATING SYSTEM IS RESPONSIBLE FOR THE FOLLOWING ACTIVITIES IN CONNECTION WITH MEMORY MANAGEMENT:
  - ✓ KEEPING TRACK OF WHICH PARTS OF MEMORY ARE CURRENTLY BEING USED AND BY WHOM
  - ✓ DECIDING WHICH PROCESSES (OR PARTS) AND DATA TO MOVE INTO AND OUT OF MEMORY
  - ✓ ALLOCATING AND DE-ALLOCATING MEMORY SPACE AS NEEDED
- THE CONCEPT OF A **PROCESS LOGICAL ADDRESS SPACE :**
  - ✓ SET OF ADDRESSES ALLOCATED TO A SINGLE PROCESS
  - ✓ GENERATED BY THE CPU
- **PHYSICAL ADDRESSES –** ACTUAL LOCATION IN MAIN MEMORY

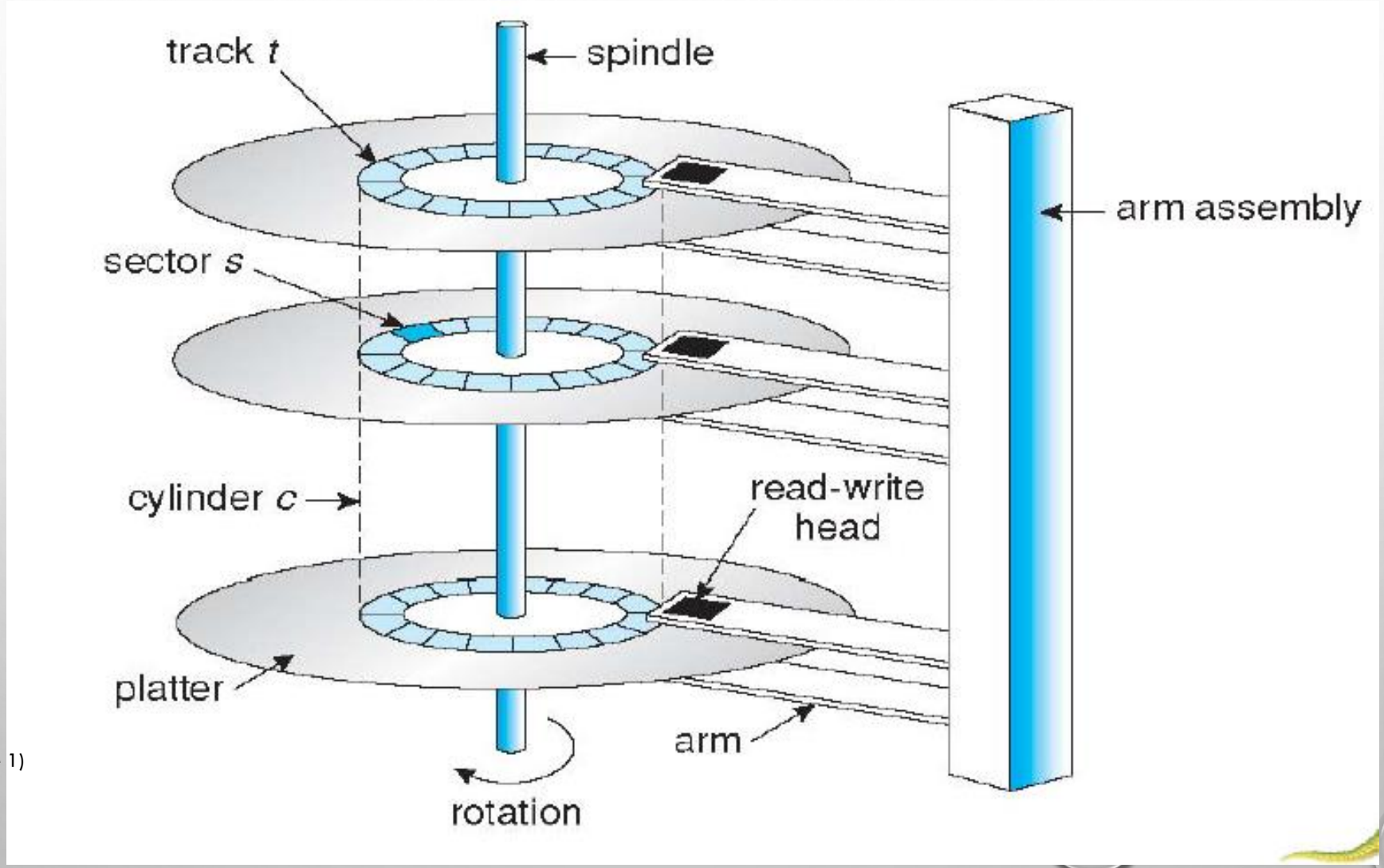
VIRTUAL MEMORY



I/O SUBSYSTEM: MASS STORAGE

- MAGNETIC DISKS PROVIDE BULK OF SECONDARY STORAGE OF MODERN COMPUTERS DRIVES
- DISK SYSTEM FUNCTIONING IS BASED ON TWO MAIN IDEAS:
  - MAGNETIC RECORDING
  - FAST DISK PLATTERS ROTATION
- DISKS CAN BE REMOVABLE
- DRIVE ATTACHED TO COMPUTER VIA I/O BUS
- BUSSES VARY, INCLUDING EIDE, ATA, SATA, USB, FIBRE CHANNEL, SCSI, SAS, FIREWIRE

DISK DRIVE SYSTEM



DISK FORMAT

- DISK FORMAT – C/H/S (CYLINDERS/HEADS/SECTORS)
- **A DISK TRACK** IS A CIRCULAR PATH ON THE SURFACE OF A DISK (PLATTER) ON WHICH INFORMATION IS MAGNETICALLY RECORDED AND FROM WHICH RECORDED INFORMATION IS READ (OR HEAD TRAJECTORY)
- **A CYLINDER** – IS A SET OF ALL TRACKS ON ALL SURFACES AND ALL PLATTERS OF A DISK SYSTEM HAVING THE SAME DIAMETER
- THE NUMBER OF TRACKS ON A SINGLE SURFACE IN THE DRIVE EXACTLY EQUALS TO THE NUMBER OF CYLINDERS OF THE DRIVE
- A **SECTOR** IS A SUBDIVISION OF A TRACK
  - ✓ THE MINIMUM STORAGE UNIT OF A DISK DRIVE
  - ✓ UNIT OF EXCHANGE BETWEEN DISK DRIVE AND MAIN MEMORY, ITS SIZE=512 BYTES

V (RAW DISK CAPACITY)= C*H*S*512

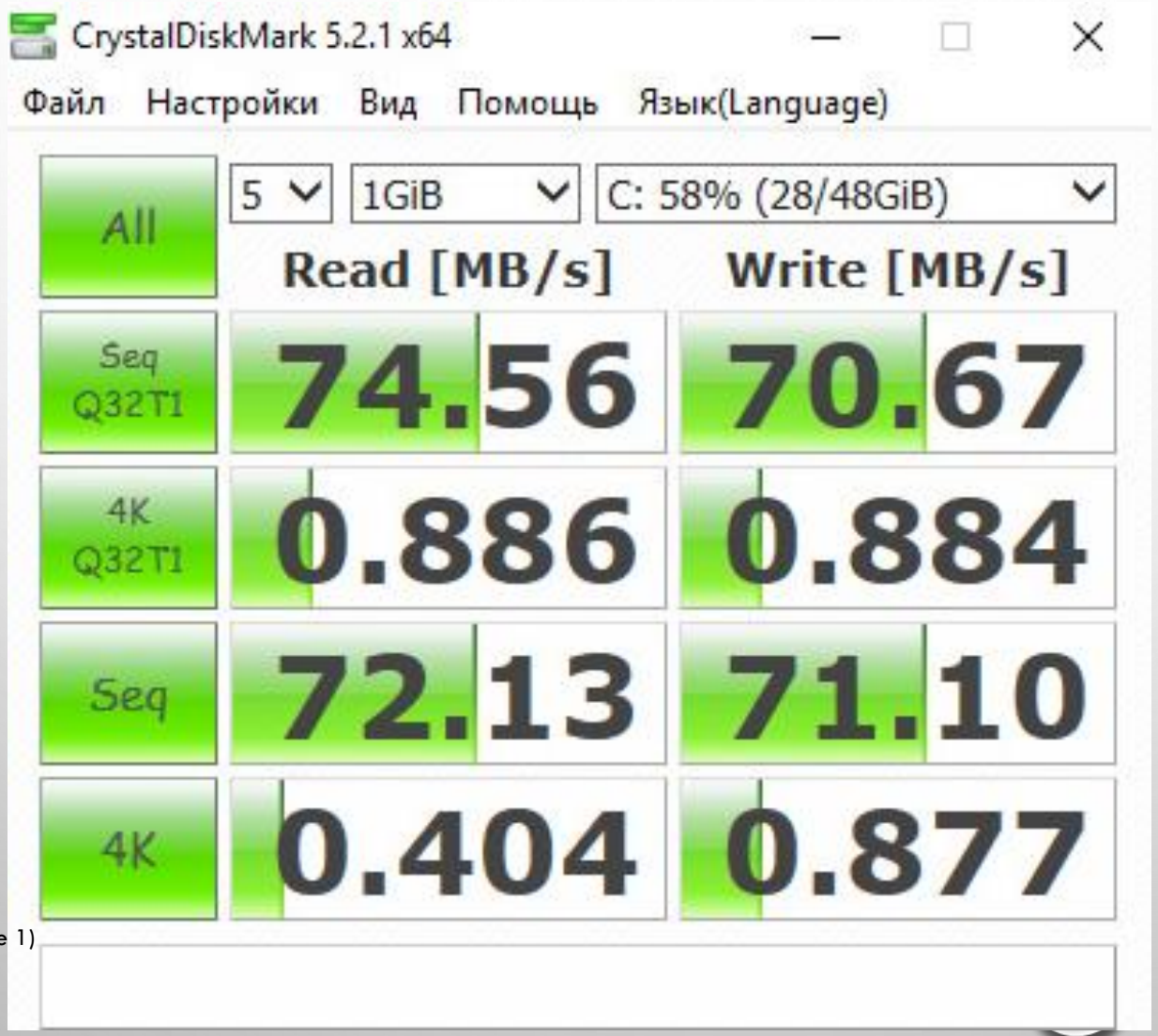
DISK DRIVE CHARACTERISTICS

- PLATTERS RANGE FROM .85” TO 14” (HISTORICALLY)COMMONLY 3.5”, 2.5”, AND 1.8”
- VOLUME RANGES FROM 30GB TO 3TB PER DRIVE
- PERFORMANCE TRANSFER RATE – THEORETICAL –6 GB/SEC (TRANSFER RATE IS THE RATE AT WHICH DATA FLOW BETWEEN DRIVE AND COMPUTER)
- EFFECTIVE TRANSFER RATE –REAL –1GB/SEC
  - LATENCY BASED ON SPINDLE SPEED 1/(RPM * 60)
- AVERAGE LATENCY = ½ LATENCY

AVERAGE LATENCY

Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

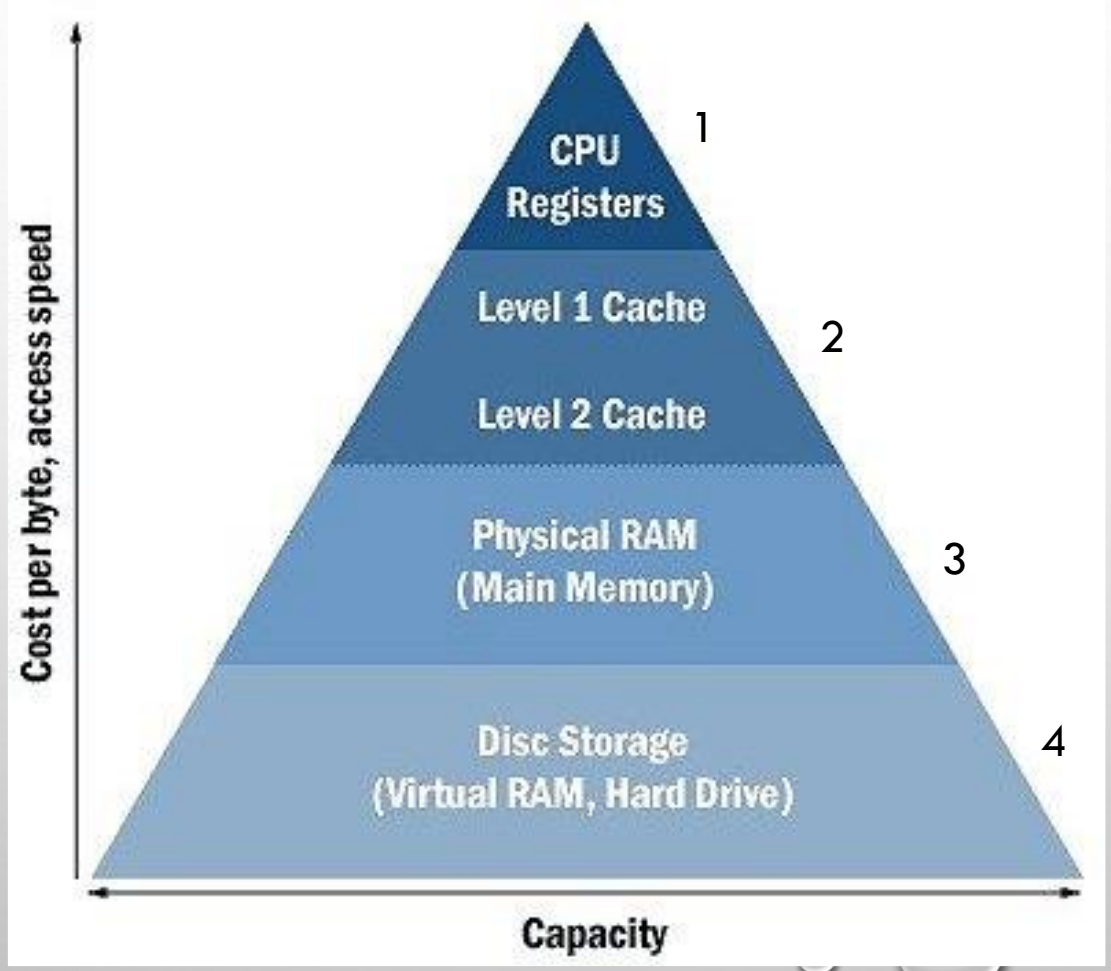
SOFTWARE FOR DISK RATES MEASURING



LOGICAL VIEW ON DISK SYSTEM

- DISK DRIVES ARE ADDRESSED AS LARGE 1-DIMENSIONAL ARRAYS OF **LOGICAL BLOCKS, WHERE THE LOGICAL BLOCK IS THE SMALLEST UNIT OF TRANSFER**
- THE 1-DIMENSIONAL ARRAY OF LOGICAL BLOCKS IS MAPPED INTO THE SECTORS OF THE DISK SEQUENTIALLY SECTOR 0 IS THE FIRST SECTOR OF THE FIRST TRACK ON THE OUTERMOST CYLINDER
- MAPPING PROCEEDS IN ORDER THROUGH THAT TRACK, THEN THE REST OF THE TRACKS IN THAT CYLINDER, AND THEN THROUGH THE REST OF THE CYLINDERS FROM OUTERMOST TO INNERMOST

MEMORY/STORAGE STRUCTURE



PERFORMANCE OF DIFFERENT LEVELS MEMORY

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	< 16 MB	< 64 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system