

# OPERATING SYSTEMS

## operating systems

### what is it ?

DEFINITION: THE KERNEL AND ANY PROGRAMS NECESSARY TO PROVIDE SERVICES TO APPLICATIONS

- \* MANAGES HARDWARE RESOURCES AMONG COMPETING ENTITIES
- \* PROVIDES COMMON SERVICES TO APPLICATIONS
- \* INTERFACES APPLICATIONS TO HARDWARE



### resource manager

- \* PROTECTS ENTITIES FROM INTERFERING WITH EACH OTHER
- \* KEEPS EVERYTHING WHERE IT NEEDS TO BE
- \* PROTECTS ITSELF AND OVERALL SYSTEM FROM MALICIOUS AND ACCIDENTAL DAMAGE

## Service provider

- \* PROVIDES COMMON FUNCTIONALITY → SIMPLIFIES APPLICATIONS
- \* PROVIDES STANDARD INTERFACES, MAKING APPLICATIONS MORE CONSISTENT
- \* THE "COMMON CORE" OF OS

## Virtual machine

- \* DEFINES A NUMBER OF CHARACTERISTICS NOT FOUND ON THE HARDWARE
- \* PRESENTS A CLEANER AND EASIER MACHINE FOR APPLICATIONS TO RUN ON
- \* HIDES THE BOUNDARY BETWEEN OS & HARDWARE ALLOWING FLEXIBILITY IN HARDWARE DESIGN WHILE STILL RUNNING THE SAME PROGRAMS
- \* RUN A VIRTUAL MACHINE TO RUN PROGRAMS NOT ON YOUR COMPUTER

# responsibilities

## \* HARDWARE

→ CPU

→ INPUT/OUTPUT DEVICES

- \* PROVIDES COMMON DEVICE INTERFACE TASKS

- \* MANAGES EXCLUSIVE ACCESS

- \* HIDES DEVICE DETAILS

- \* ESTABLISH ACCESS TO A DEVICE

- \* RELEASE A DEVICE

- \* READ FROM A DEVICE

- \* WRITE TO A DEVICE

- \* GRANT ACCESS TO A DEVICE

- \* PROVIDE SPECIAL DEVICE OPERATIONS

→ MEMORY

- \* ASSIGNS AREAS OF MEMORY BELONGING TO PROCESSES TO AREAS OF PHYSICAL MEMORY

- \* MANAGES REQUESTS THAT EXCEED AVAILABLE MEMORY

- \* CONTROLS SHARING OF MEMORY

- \* DIRECT ALLOCATION REQUESTS

- \* DIRECT FREING OF MEMORY

- \* SERVING MEMORY NEEDS IMPLICIT IN OTHER SERVICES

- \* MANAGING SHARING OF AREAS OF MEMORY

## \* FILE SYSTEMS

- PROVIDES TRANSLATION FROM NAMES TO RESOURCES
- MANAGES PERSISTENT STORAGE OF DATA
- PRESENTS A DEVICE-INDEPENDENT STORAGE
- PROTECTS RESOURCES FROM UNAUTHORIZED ACCESS
- SERVICES

\* OPEN A FILE      \* CLOSE A FILE

\* READ FROM A FILE      \* WRITE TO A FILE

\* SEEK IN A FILE      \* QUERY & MODIFY FILE PARAMETERS

## \* SECURITY

- PART OF OTHER MANAGEMENT RESPONSIBILITIES
- AUTHENTICATES THE IDENTITY OF A REQUESTER
- AUTHORIZES ACCESS ACCORDING TO A SECURITY POLICY

## \* OTHER AREAS

— NETWORKING

\* IMPLEMENTS PROTOCOL STACKS

\* PROVIDES SERVICES FOR

- ESTABLISHING CONNECTIONS TO REMOTE SYSTEMS
- LISTENING FOR CONNECTIONS FROM REMOTE SYSTEMS
- EXCHANGING DATA WITH REMOTE SYSTEMS

— USER INTERFACES

— PROCESSES ON NEXT PAGE

## — PROCESSES

- \* MANAGES CPU IN TERMS OF RUNNING PROCESSES CALLED PROGRAMS
- \* SCHEDULES PROCESSES, PICKING THE NEXT ONE TO GET THE CPU
- \* SWITCHES BETWEEN PROCESSES, CALLED CONTEXT SWITCHING
- \* CREATING PROCESSES
- \* TERMINATING PROCESSES
- \* CHANGING PROCESS PARAMETERS
  - eg - PRIORITY
- \* PROVIDING INTERPROCESS COMMUNICATION
- \* PROVIDING PROCESS SYNCHRONIZATION

## Early OS History

- \* NO OS ON EARLIEST MACHINES
- \* COLLECTIONS OF USEFUL ROUTINES
  - FIRST CODE REUSE
  - FORESHADOWED THE SERVICE PROVIDER ASPECT OF OS
- \* MACHINES SCHEDULED WITH SIGN UP SHEETS

## Batch OS

- \* MOST CPU TIME WASTED DURING TIME SLOTS
- \* SEPARATE USERS FROM THE CPU FOR MORE EFFICIENT CPU USAGE
- \* TAKE PROGRAMS, RUN THEM TO COMPLETION + PRODUCE RESULTS
- \* TRADE OFF EFFICIENT CPU USAGE FOR LOSS OF USER EFFECTIVENESS
- \* POTENTIALLY LONG TIME BETWEEN SUBMITTING A JOB + GETTING RESULTS

## Time sharing OS

- \* SHORTER TIME SLOTS MEAN MORE EFFICIENT CPU USAGE
- \* TAKE SHORT TIME SLOTS TO THE LIMIT
- \* RAPIDLY SWITCH AMONG RUNNING PROGRAMS AND PROVIDE MULTIPLE USER INTERFACES
- \* USEFUL EVEN WITH A SINGLE USER



## Distributed OS

- AFTER ALLOWING MULTIPLE USERS + PROGRAMS, MULTIPLE MACHINES COME NEXT
- MANAGE MULTIPLE COMPUTING SYSTEMS AS A SINGLE POOL OF COMPUTING RESOURCES
- SHARE FILE SYSTEMS
- SUPPORT MULTIPLE CPUs COOPERATING ON A SINGLE TASK

## Key Trends

- \* MOVE AWAY FROM ONE USER, ONE PROGRAM, ONE COMPUTER
  - BREAK THE TRIANGLE OF ONES



- \* MAKE INTERFACES MORE UNIFORM
- \* TRADE OFF CPU CYCLES FOR USER CONVENIENCE
  - CPU TIME GETS LESS EXPENSIVE
  - HUMAN TIME GETS MORE EXPENSIVE

# OS structure

## \* MONOLITHIC

- KERNEL STRUCTURED AS A SINGLE PROGRAM
- USUAL DESIGN TECHNIQUES
- OFTEN CRITICIZED

## \* LAYERED DESIGN

- DEPENDENCY ON ONLY LOWER LAYERS
- TRICKY TO STRUCTURE
- OFTEN REQUIRE SPLITTING SUBSYSTEMS INTO MULTIPLE LAYERS
- LAYERED EXAMPLE

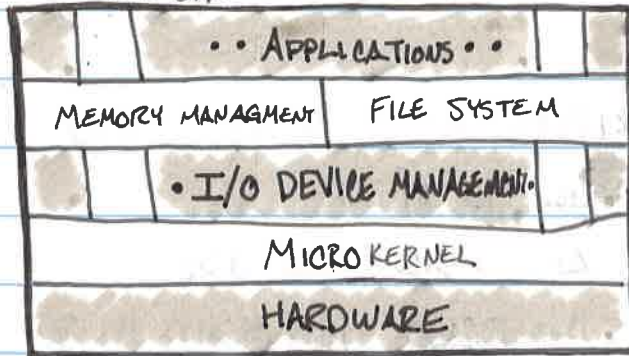


## \* MICRO KERNELS

- MOVE MUCH OF THE TRADITIONAL KERNEL FUNCTIONALITY INTO SEPARATE PROGRAMS
- \* REDUCE SIZE OF KERNELS
- \* COMPONENTS COMMUNICATE WITH MESSAGES PASSED THROUGH THE MICRO KERNEL



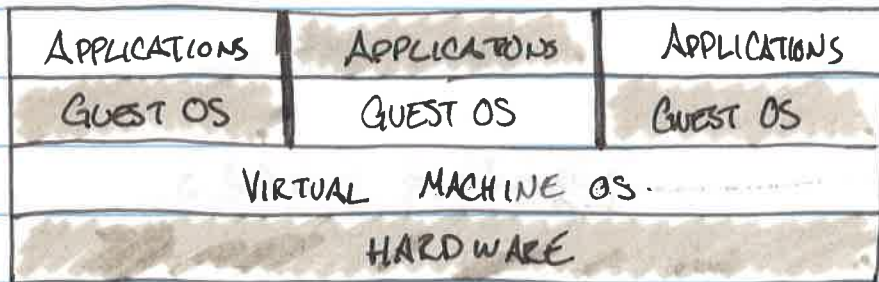
## MICROKERNEL EXAMPLE



## virtual machine OS

### \* VIRTUALIZE THE HARDWARE

- CREATES ILLUSION OF MULTIPLE SCREENS
- EACH GUEST BELIEVES IT HAS THE HARDWARE TO ITSELF
- ALLOWS DIFFERENT OSs TO RUN AT THE SAME TIME



## bootstrapping

- \* LOAD KERNEL INTO MEMORY AND TRANSFER CONTROL TO IT

- \* LOADERS ARE LIKE MINI OSs

  - UNDERSTAND MEM MGMT TO PUT KERNEL INTO MEM

  - UNDERSTAND I/O and FILE SYSTEMS TO LOCATE KERNEL AND READ IT INTO MEM

- \* PC BOOTSTRAPPING SEQUENCE

- 1) BIOS LOADS MASTER BOOT RECORD (MBR) FROM FIRST BLOCK ON THE DISK

- 2) MBR LOADS FIRST BLOCK FROM ACTIVE PARTITION

- 3) PARTITION BOOT BLOCK:

  - \* SOME LOAD KERNEL DIRECTLY

  - \* SOME LOAD SECONDARY LOADER WHICH LOADS KERNEL

## system calls

- \* INTERFACE BETWEEN APPS AND KERNELS

- \* LOOKS LIKE FUNCTION CALLS

- \* USUALLY IMPLEMENTED WITH A SOFTWARE INTERRUPT

- \* PROCESSED IN A HIGHER PRIVILEGE PROCESSOR MODE

# SOFTWARE

SYSTEM	vs APPLICATION
* SOFTWARE CONTROLLING AND MANAGING THE HARDWARE	* NON-SYSTEM SOFTWARE * END-USER TASKS
EG: OPERATING SYSTEM	EG: WORD, EXCEL, GAMES, PAYROLL

# DATABASE

## \* FUNCTIONS

- TRANSACTION PROCESSING
- QUERIES
- BATCH PROCESSING
- REPORT WRITING
- DATA ANALYSIS

## \* TYPES

- FLAT FILES
- RELATIONAL
- HIERARCHICAL
- OBJECT