OPERATING SYSTEMS

Introduction and Process Management

Venkatesh Prasad
Department of Computer Science

OPERATING SYSTEMS

Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

OPERATING SYSTEMS

Named and Unnamed Pipes

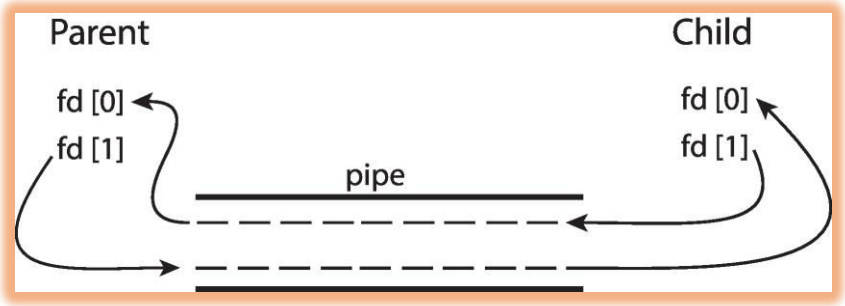
Venkatesh Prasad
Department of Computer Science

Pipes

- Acts as a conduit allowing two processes to communicate
- Issues:
  - Is communication unidirectional or bidirectional?
  - In the case of two-way communication, is it half or full-duplex?
  - Must there exist a relationship (i.e., ***parent-child***) between the communicating processes?
  - Can the pipes be used over a network?
- Ordinary (Unnamed) pipes – cannot be accessed from outside the process that created it. Typically, a parent process creates a pipe and uses it to communicate with a child process that it created.
- Named pipes – can be accessed without a parent-child relationship.

Ordinary Pipes

- Ordinary Pipes allow communication in standard producer-consumer style
- Producer writes to one end (the **write-end** of the pipe)
- Consumer reads from the other end (the **read-end** of the pipe)
- Ordinary pipes are therefore unidirectional
- Require parent-child relationship between communicating processes



- Windows calls these **anonymous pipes**
- Used by the shell; not used very often by application programs
  - Main limitation is processes need to be related

Named Pipes

- Named Pipes are more powerful than ordinary pipes
- Communication is bidirectional
- No parent-child relationship is necessary between the communicating processes
- Several processes can use the named pipe for communication
- Provided on both UNIX and Windows systems.
- Limitations of named pipes:
  - Two pipes needed for two way communication
  - FIFO: Data once retrieved is removed from pipe

Named Pipes (Cont.)

- On UNIX Systems:
  - Named Pipes are referred to as FIFOs
  - appear as typical files in the file system
  - Created with the system call mkfifo() and manipulated with open(), read(), write() and close() system calls
  - Only half-duplex transmission is permitted; 2 FIFOs are used for data travel in both directions
  - Allows only byte-oriented data transmission across a FIFO
  - Communicating processes must reside on the same machine

Named Pipes (Cont.)

- On Windows systems:
  - Created with the function CreateNamedPipe() and manipulated with ConnectNamedPipe(), ReadFile(), WriteFile() and DisconnectNamedPipe() functions
  - Full duplex transmission is permitted
  - Communicating processes may reside on the same or different machines
  - Allows byte- or message-oriented data transmission

THANK YOU

Venkatesh Prasad
Department of Computer Science Engineering

venkateshprasad@pes.edu