Virtual Machine Basics

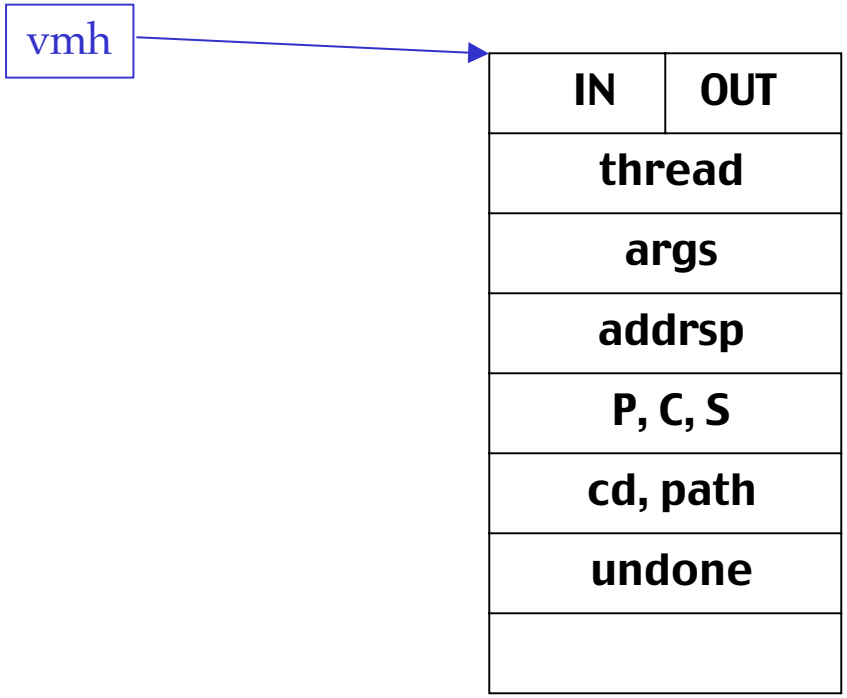
P. J. Denning

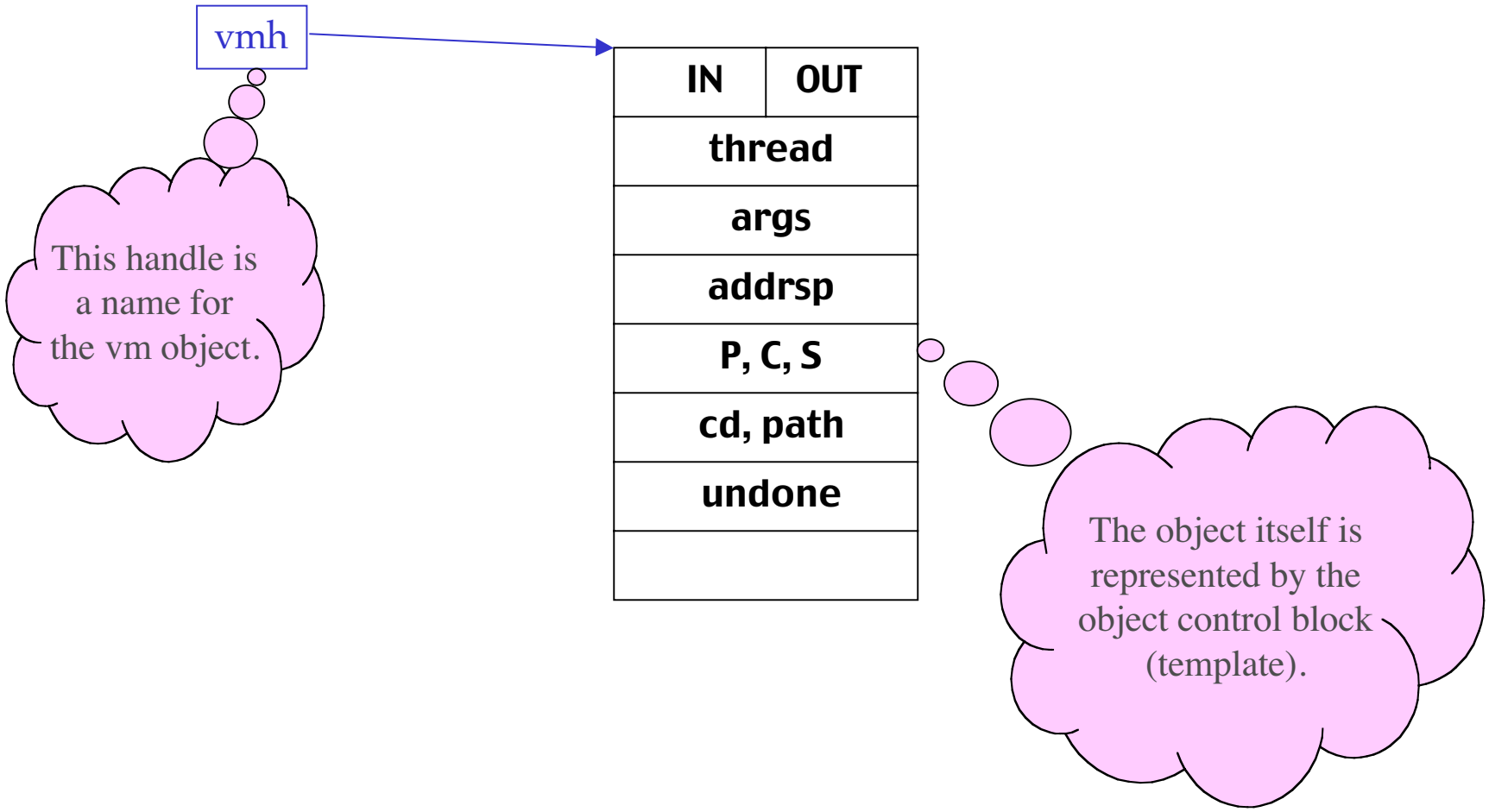
For CS471/CS571

- Four meanings of "virtual machines"
  - Exact replica of hardware within a partition of memory (IBM VMS)
  - Simulation of one machine on another (Virtual PC)
  - Abstract machines (level structured OS)
  - Standard form for program-in-execution (Unix)
- We focus on the fourth meaning.

What is a virtual machine?

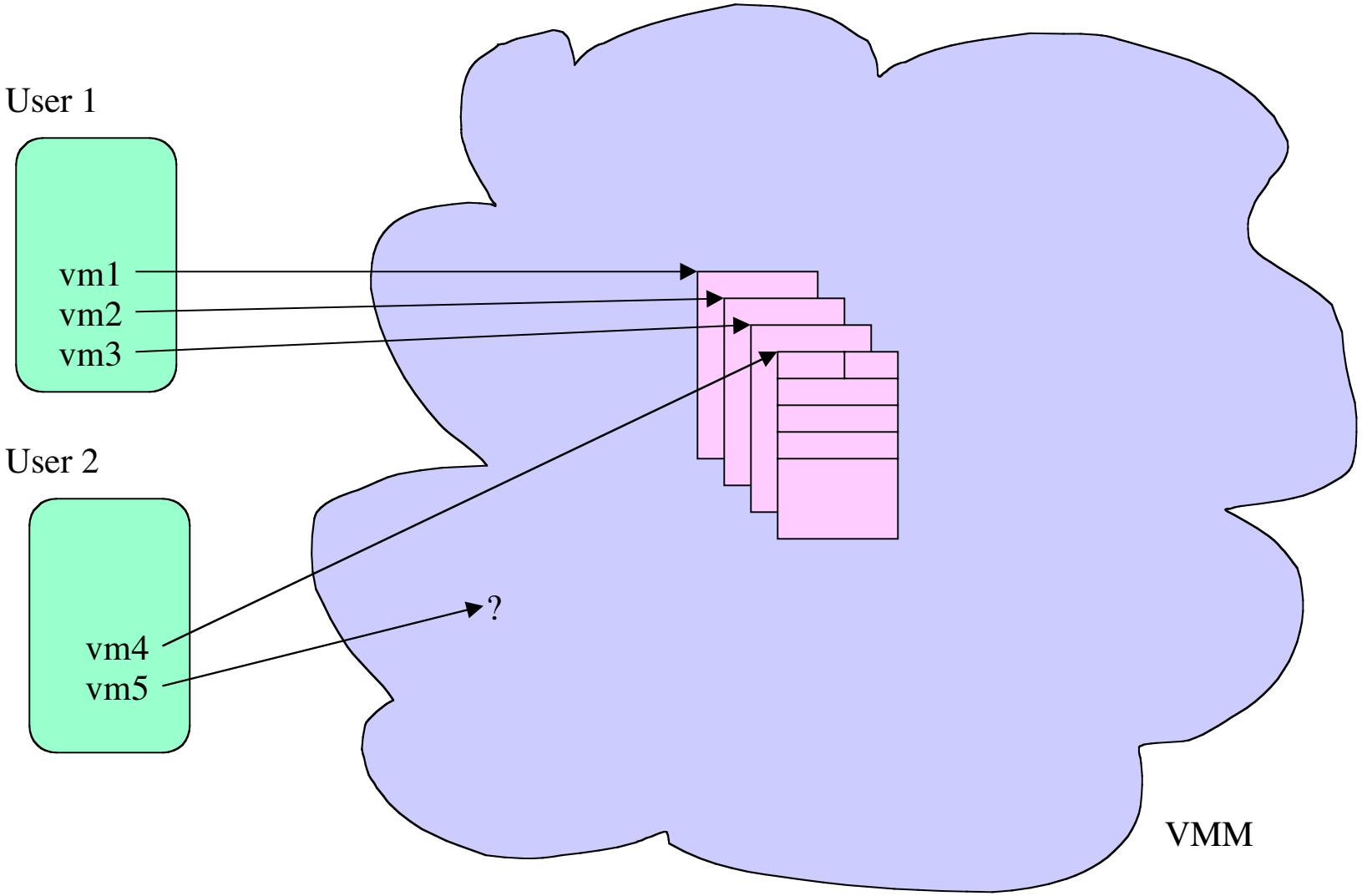
- A simulated computational environment for a program, consisting of
  - An IN and OUT port
  - A thread running a program
  - A list of arguments (args)
  - An address space holding code, data, stack
  - Parent, children, and sibling pointers
  - A current directory pointer (and path)
  - A count of undone children





Virtual Machine Manager

- A subsystem that manages all VMs
- System gives each VM a ***handle*** on creation
- Handles stored in user workspace
- Users present handles (to VMM) with requests to perform operations on the VMs named by the handles
- Only the external operations offered by the VMM can be invoked



Operations

- CREATE

  vmh = create_vm(specs for all fields)
- DELETE

  delete_vm(vmh)
- COMPUTE

  compute
- EXIT

  exit

Execution Model

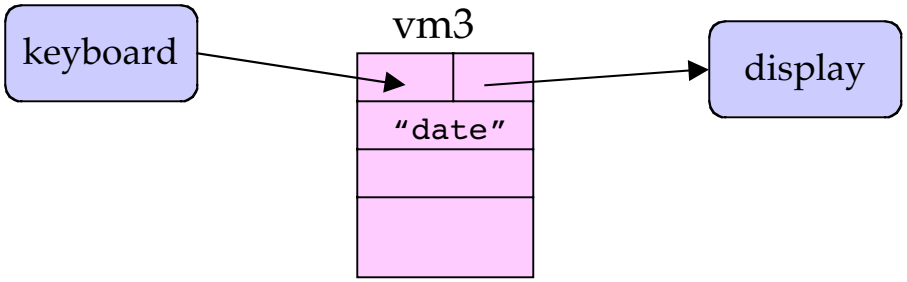
- Parent creates one or more children VMs, all initially in a wait state
- Parent interconnects VMs with pipes or attaches open files to their IN/OUT ports
- Parent issues COMPUTE command, which starts all children, sets UNDONE = children count, and puts parent to sleep
- Each child completes with EXIT, which deducts 1 from UNDONE count of parent and awakens parent when UNDONE = 0

- This model ensures that parent and children (as group) are mutually excluded in their executions
- No race conditions between children and parent for use of parent's standard IN and OUT

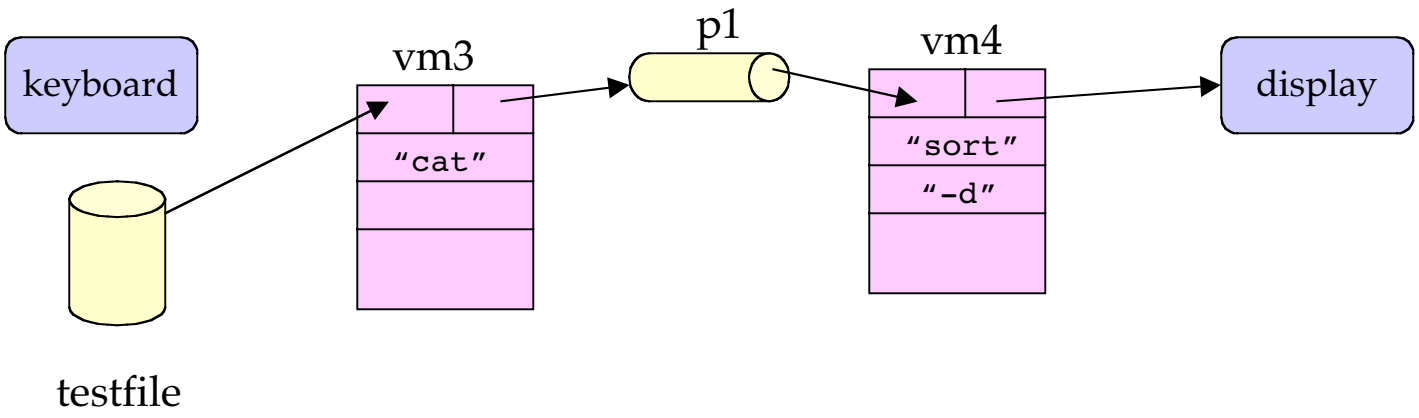
Command Interpreter (Shell)

- Listen for user to type line of text
- Decompose text into substrings, each corresponding to command component (parser)
- Create action script that tells OS how to create a VM structure to execute the command (parser)
- With COMPUTE command initiate the VM structure; wait for all its children processes to EXIT
- Clean up
- Generate command prompt and repeat

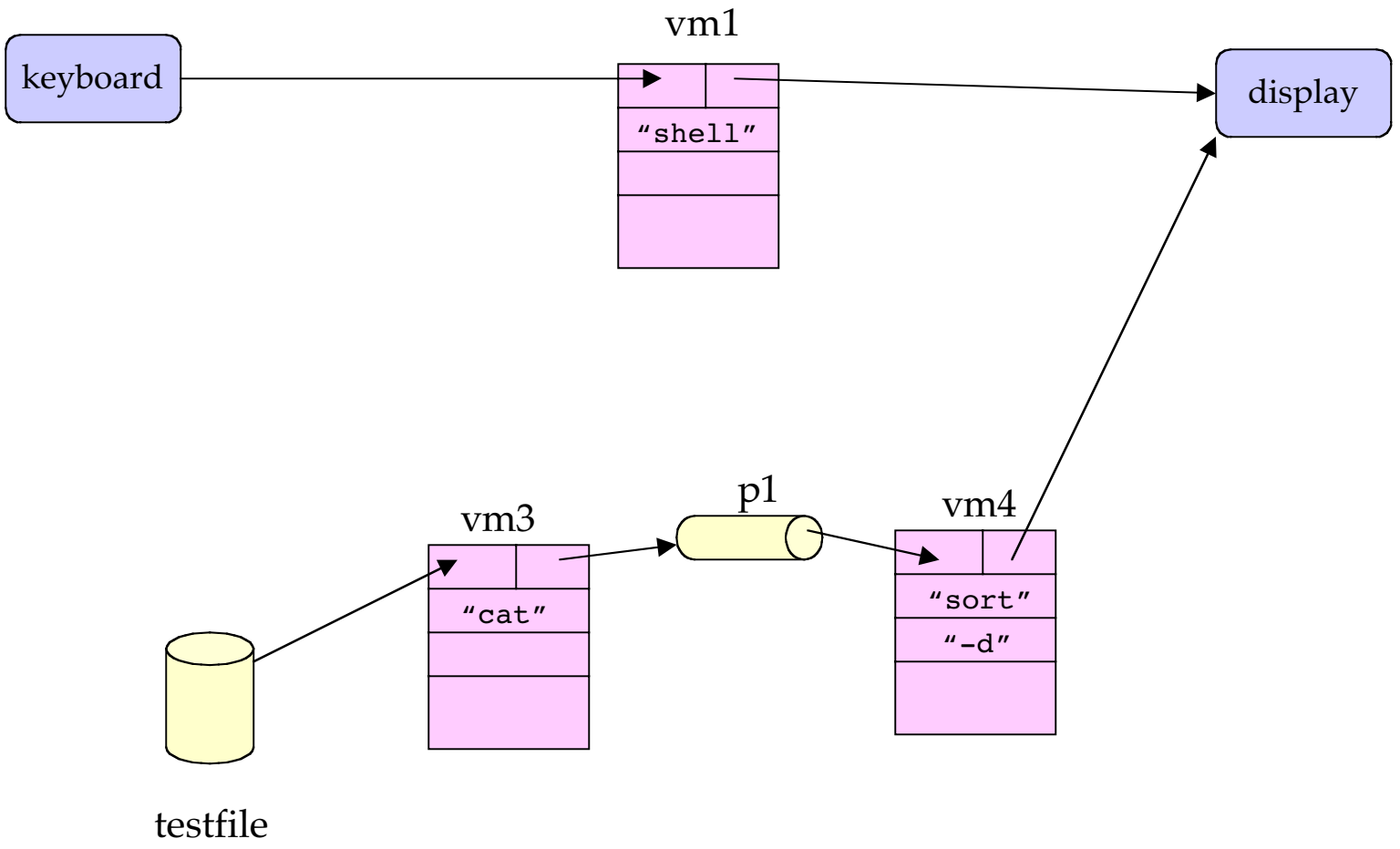
date



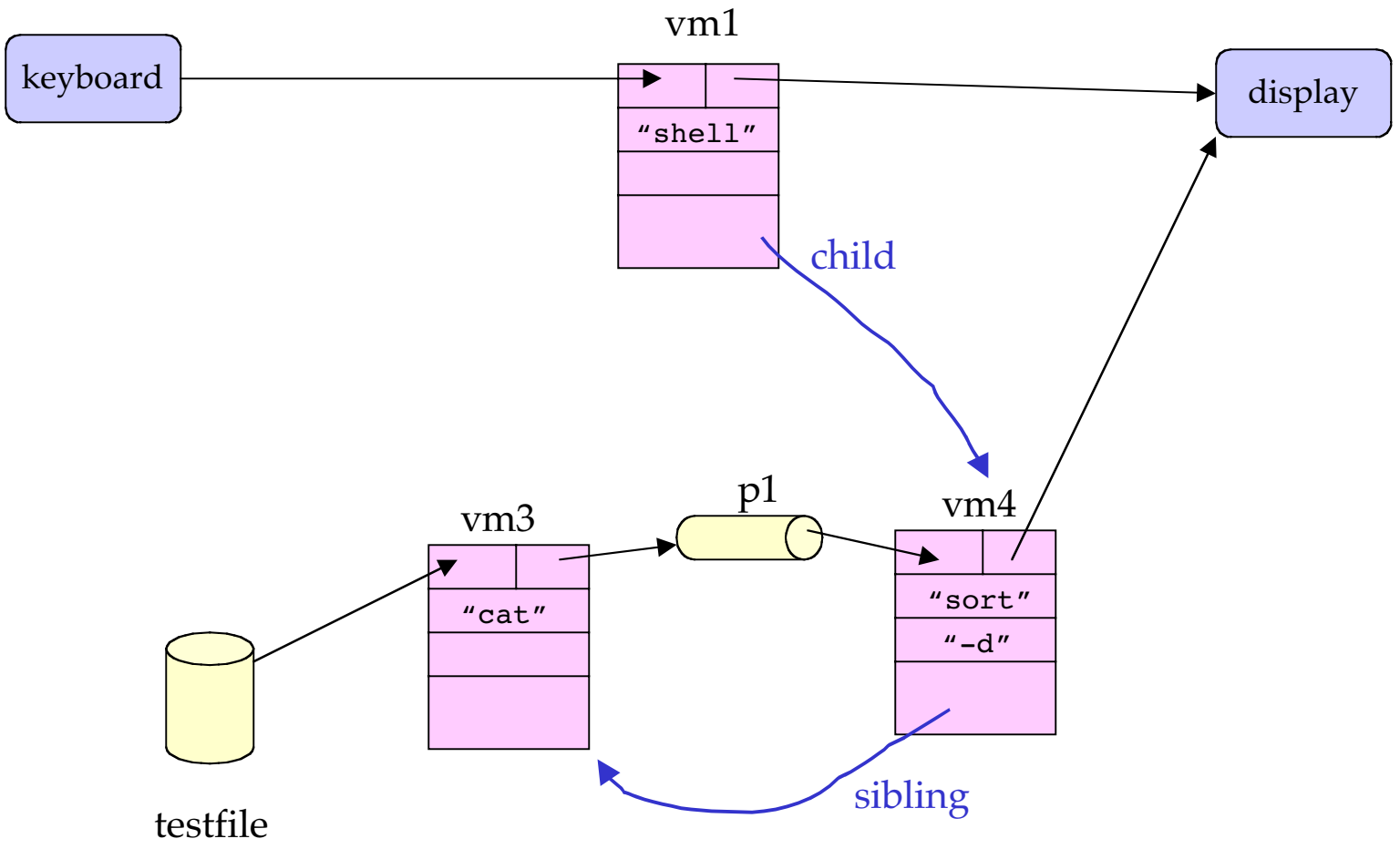
```
cat < testfile | sort -d
```



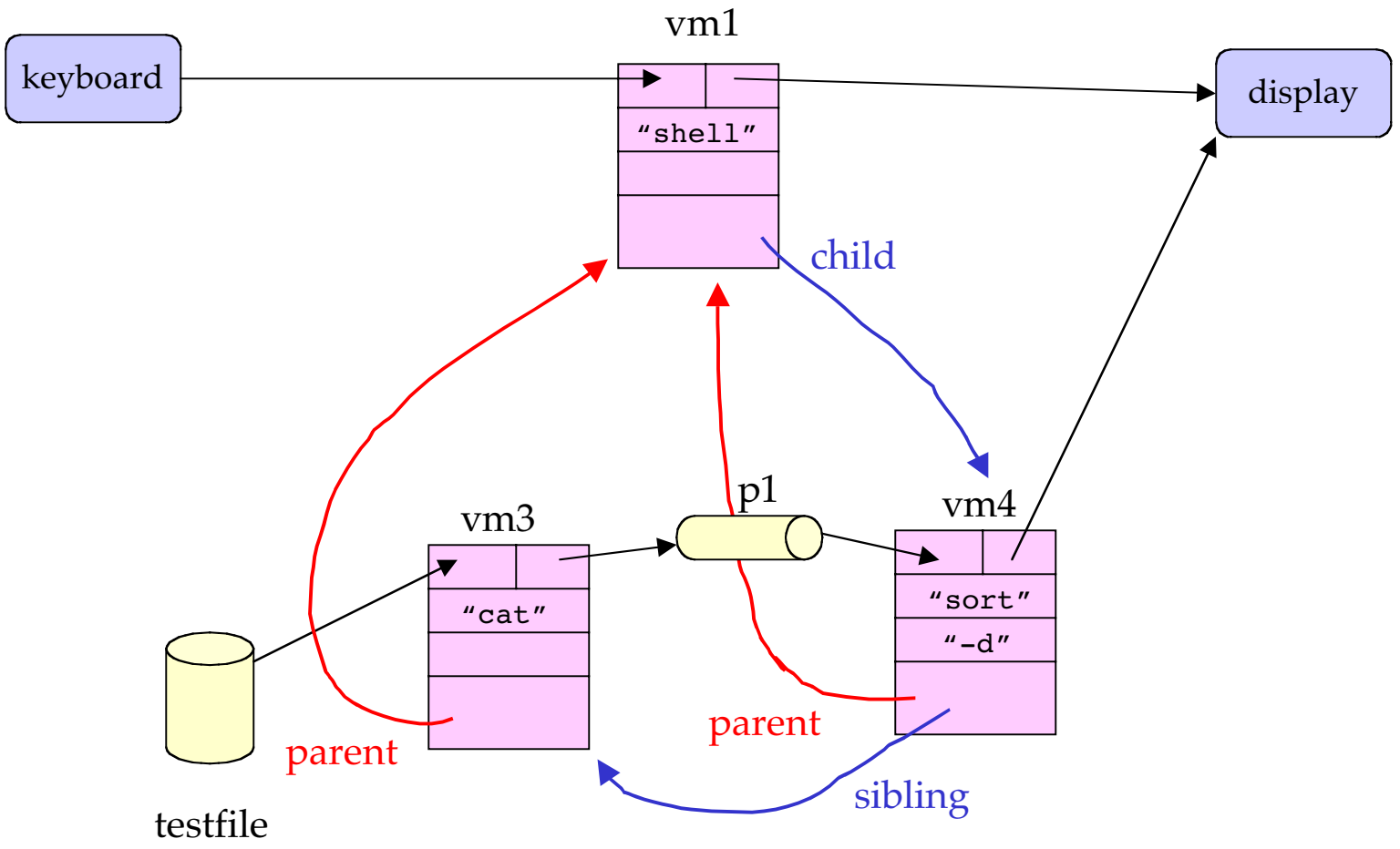
```
cat < testfile | sort -d
```

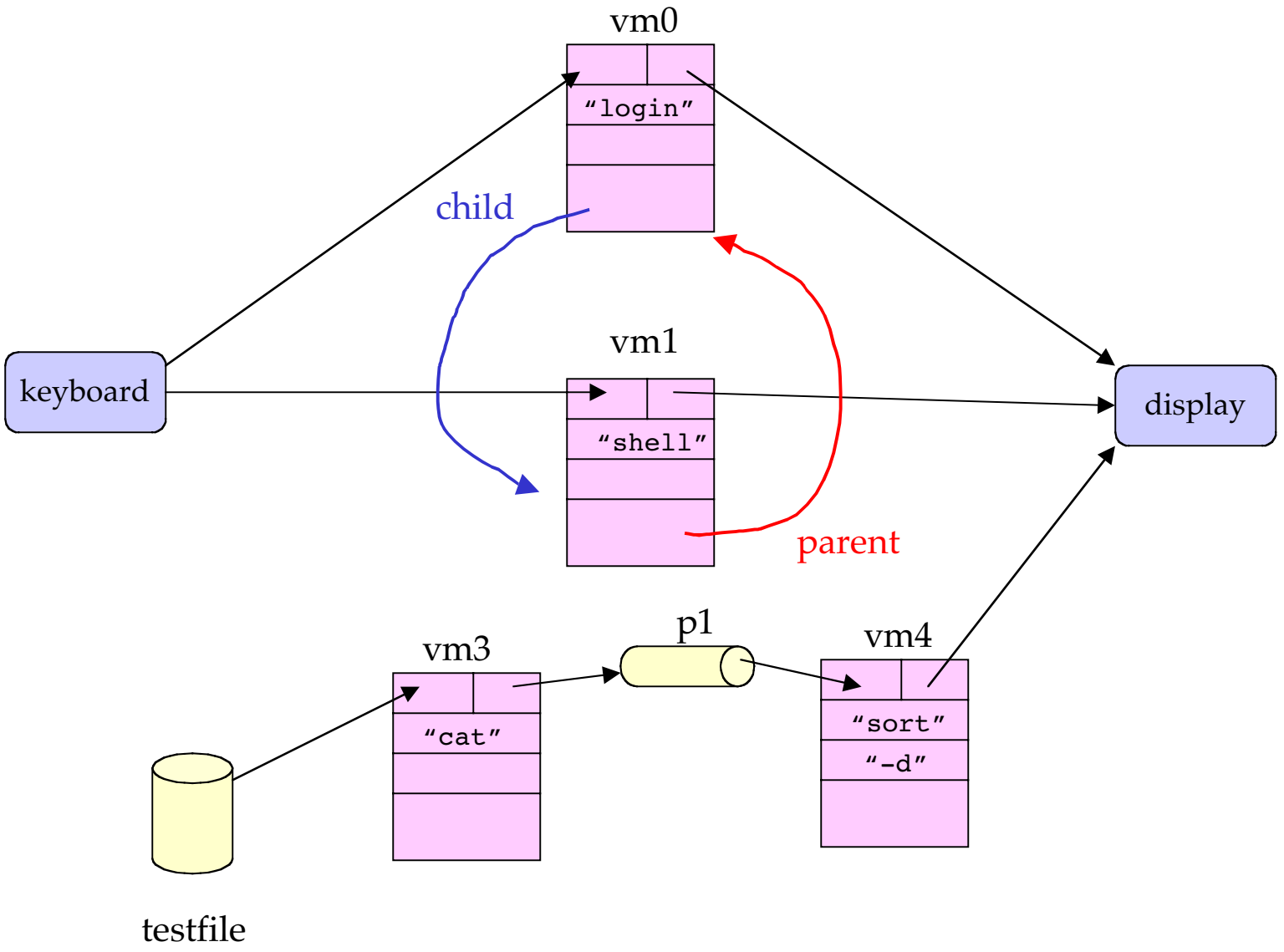


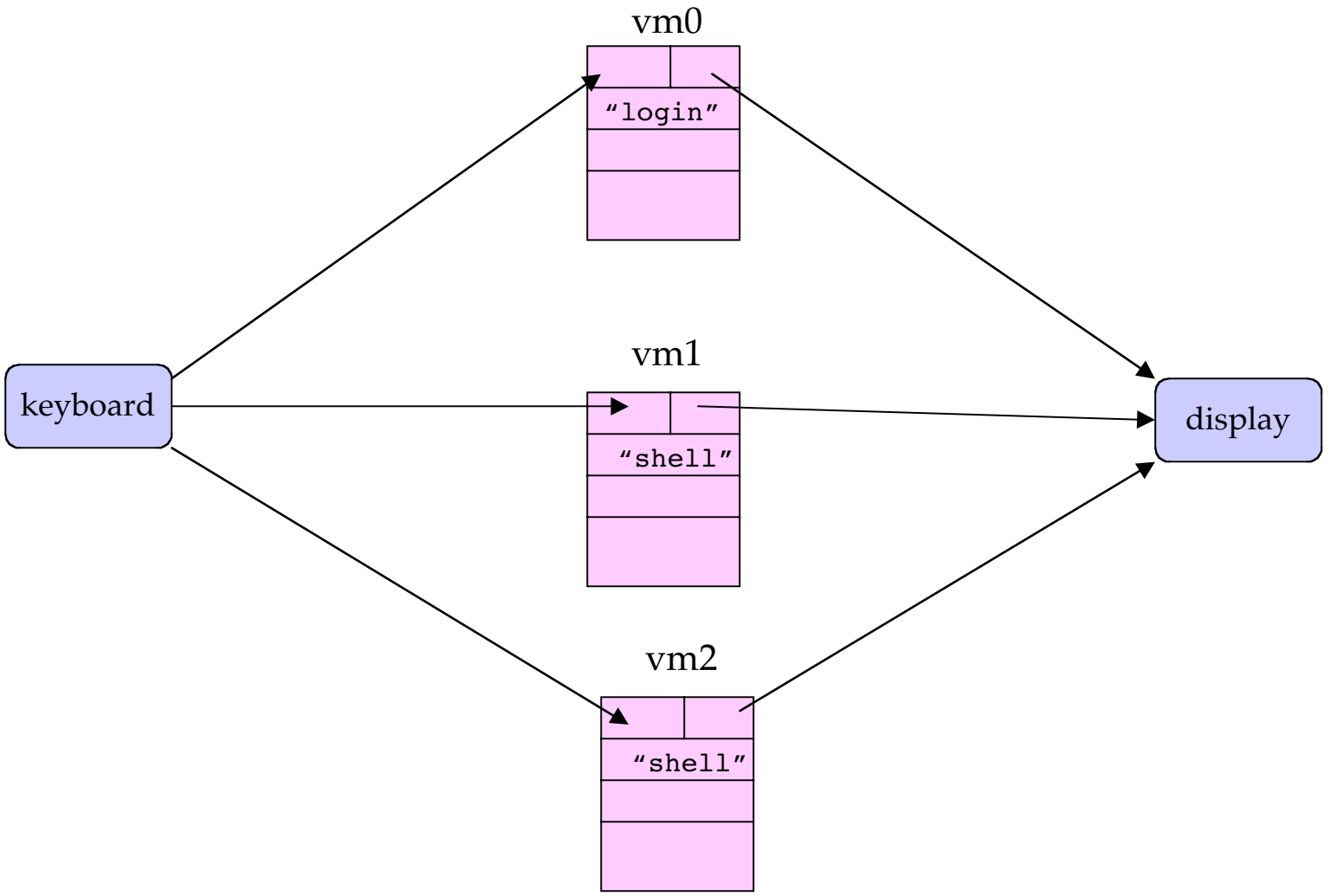
```
cat < testfile | sort -d
```



```
cat < testfile | sort -d
```







Recursive Shell

- "shell" is an allowable command --- creates a new shell VM as child to the issuer
- "login" is an allowable command; logs out old session (entire subtree from the login) and starts over

Parser

- Component of command interpreter.
- Scans text strings to identify syntactic components according to a grammar
- Issues actions that tell OS to create execution components (VMs, pipes, open files) for a command.
- See separate slide set for more detail.