Processes (Part I)

Godmar Back

Virginia Tech

August 22, 2023

VIRGINIA TECH

Processes

Definition of Process

An instance of a program that is being executed (aka a running instance)

OS provides, for each process,

- Logical flows of control:
  - 1 flow for single-threaded programs
  - multiple flows for multi-threaded programs
- Private, protected address space
- Abstracted resources (file descriptors)

These facilities abstract CPU, memory, and devices, respectively.

Processes. We consider that the system hardware comprises one or more processors, which we can identify as being distinct from the main memory, the file storage devices and the input/output devices. Each processor is capable of executing algorithms that are specified by sequences of instructions. A *process* is a locus of control within an instruction sequence. That is, a process is that abstract entity which moves through the instructions of a procedure as the procedure is executed by a processor.

Figure 1: Dennis and Van Horn: Programming Semantics for Multiprogrammed Computations, 1966 [2]

VIRGINIA TECH

Context Switching

- Historical motivation for processes was introduction of multi-programming:
  - Load multiple processes into memory, and switch to another process if current process is (momentarily) blocked, perhaps waiting for user input
  - This required protection and isolation between these processes, implemented by a privileged kernel: dual-mode operation.
- Time-sharing: a policy that switches to another process periodically to make sure all processes make progress
- The act of switching between processes is called a context switch
- "Context" here means the state of the running program, which includes the current program text, the location within the program text (PC/IP), and all associated state: variables (global, heap, stack, CPU registers)
- Because context switching is typically managed by the kernel, it interacts with mode switching

VIRGINIA TECH

Dual-Mode Operation

To enable the implementation of switching between separate contexts, processors provide the ability to operate in different "modes," or privilege levels. At a minimum, there are 2 fundamental modes:

- "kernel mode": aka system mode, supervisor or monitor mode
  - On Intel: PL0, or Privilege Level 0
- "user mode": non-privileged mode
  - On Intel: PL3, or Privilege Level 3
- These modes are maintained by the CPU (think of a bit)
- Rule 1: instructions designated as "privileged" instructions will be executed only if in kernel mode; else they cause a trap
- Rule 2: transition from/to kernel mode is carefully controlled
- Example: `HLT` instruction

VIRGINIA TECH

Mode Switching

Two directions to consider

- User → Kernel mode
  - Transitions to known, protected entry point in kernel code
  - May occur for reasons external or internal to CPU
  - External (aka hardware) interrupt
    - timer/clock chip, I/O device, network card, keyboard, mouse
    - asynchronous: unrelated to what the currently executing program does
  - Internal interrupt (aka software interrupt, trap, or exception)
    - can be intended ("trap"): for system call (process wants to enter kernel to obtain services)
    - or unintended (usually): ("fault/exception") (division by zero, attempt to execute a privileged instruction while in user mode, memory access violation, invalid instruction, alignment error, etc.)
    - synchronous: caused by what the current program does
- Kernel → User mode
  - via special privileged instruction (Intel: `iret`)
  - represents either a return from interrupt or careful action to resume user program execution

Results in Limited Direct Execution [1]

VIRGINIA TECH

Context Switch Scenarios

See Examples

VIRGINIA TECH

Context vs Mode Switches, Summary

- Mode switch guarantees that kernel gains control when needed
  - To react to external events
  - To handle error situations
  - Entry into kernel is controlled
- Not all mode switches lead to context switches
- Kernel decides if/when – subject to process state transitions and scheduling policies
- Mode switch does not change the identity of current process/thread

VIRGINIA TECH

Exceptions, Bottom-Up View

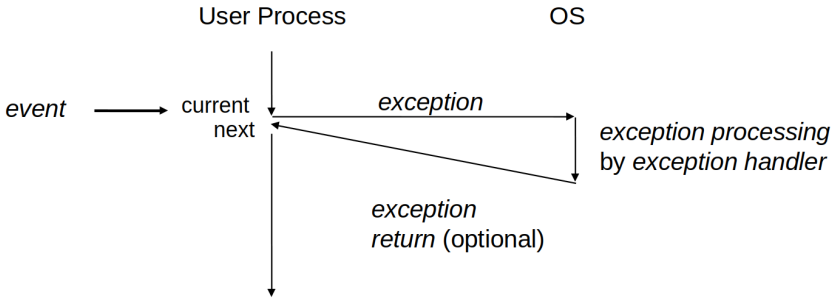

Figure 2: Figure 9.1 in CSApp3e book, "Anatomy of an Exception"

The book details exceptional control flow from the perspective of a process experiencing the exception that leads to a user → kernel mode switch.

VIRGINIA TECH

References

[1] Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau.
Operating Systems: Three Easy Pieces.
Arpaci-Dusseau Books, 1.00 edition, August 2018.

[2] Jack B. Dennis and Earl C. Van Horn.
Programming semantics for multiprogrammed computations.
Commun. ACM, 9(3):143–155, March 1966.

VIRGINIA TECH