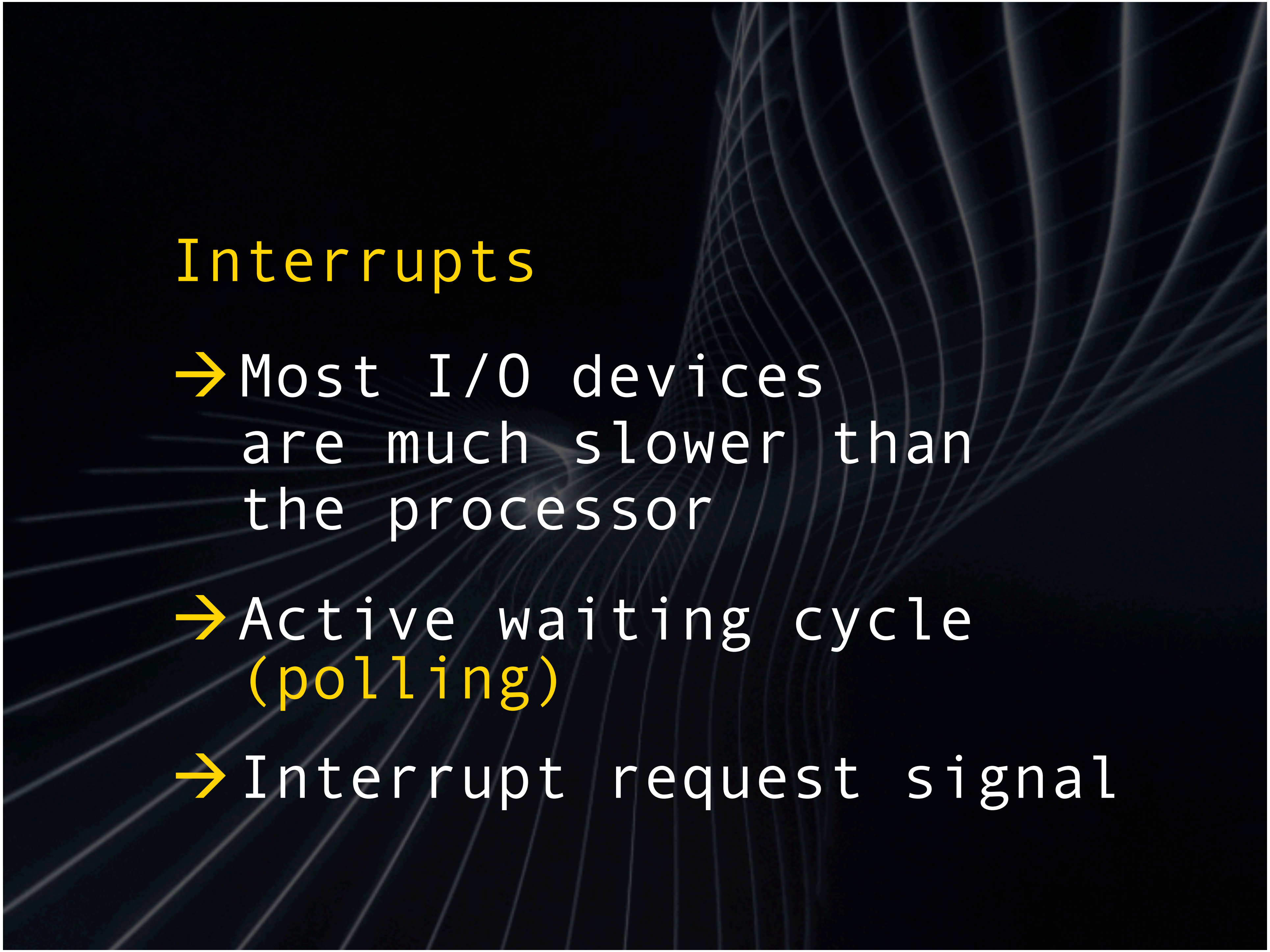
OPERATING SYSTEM OVERVIEW

Contents

Basic hardware elements



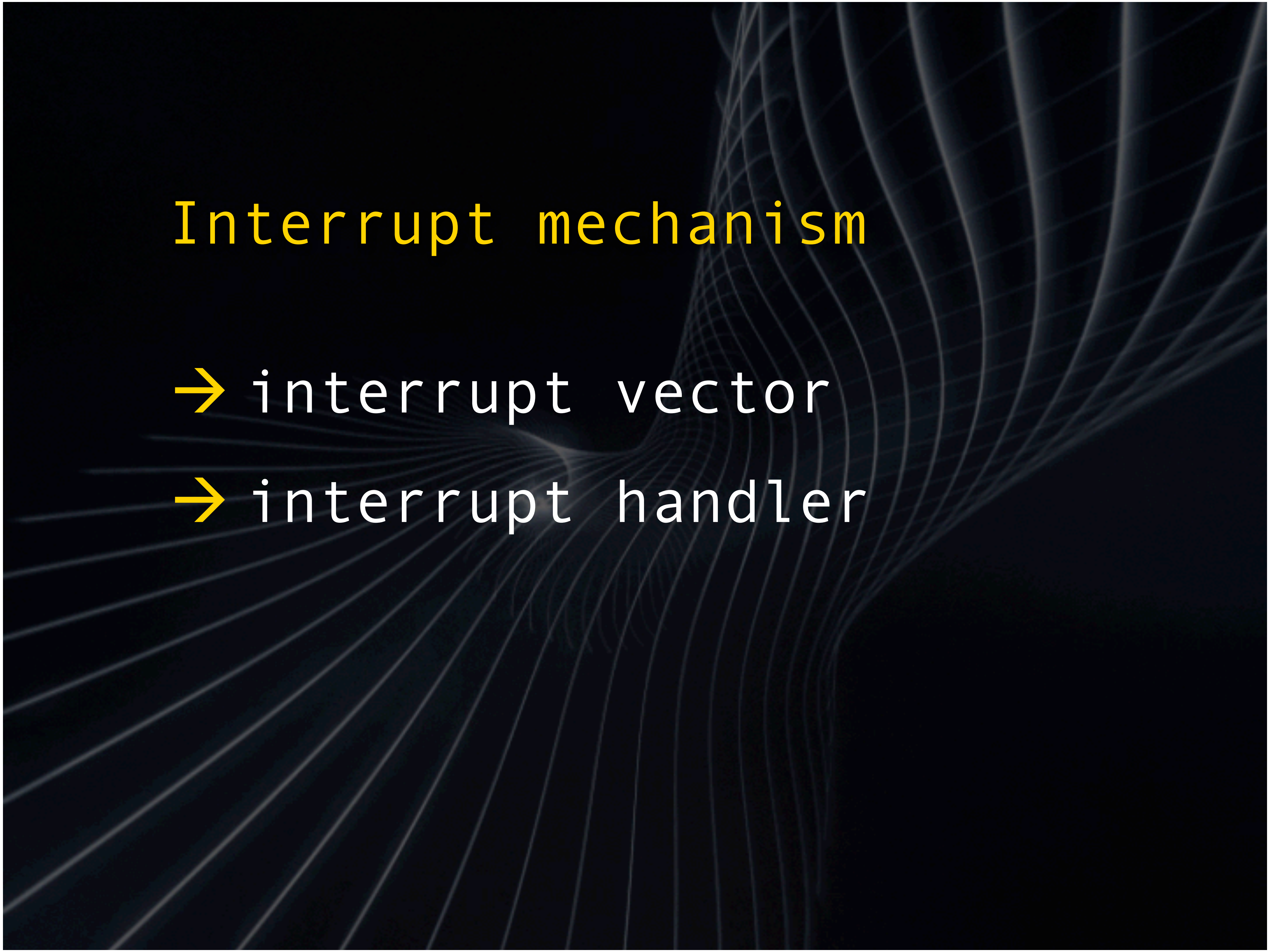
Interrupts

- → Most I/O devices are much slower than the processor
- → Active waiting cycle (polling)
- → Interrupt request signal

Interrupt mechanism

→ An interrupt cycle is added to the instruction cycle (fecth and execute).

→ The processor checks to see if an interrupt has occured (interrupt signal)



Interrupt mechanism

→ interrupt vector

→ interrupt handler

Interrupt processing

→ CPU saves PS and PC (push) into the control stack

→ CPU loads into PC and PS the corresponding values from the interrupt vector

Interrupt processing

→ interrupt handler execution

→ iret

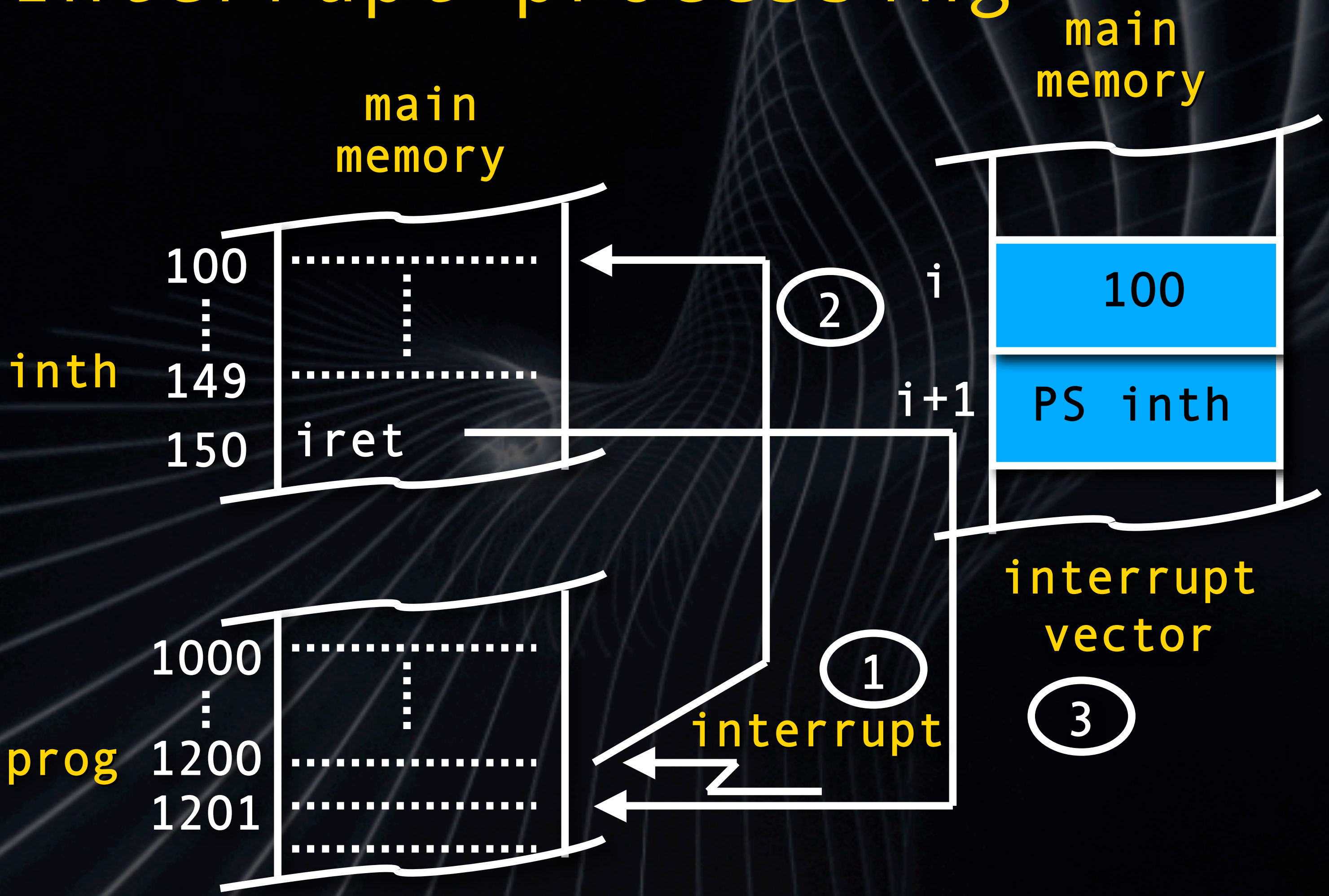
Interrupt mechanism

Interrupt enable/disable bit (PS register)

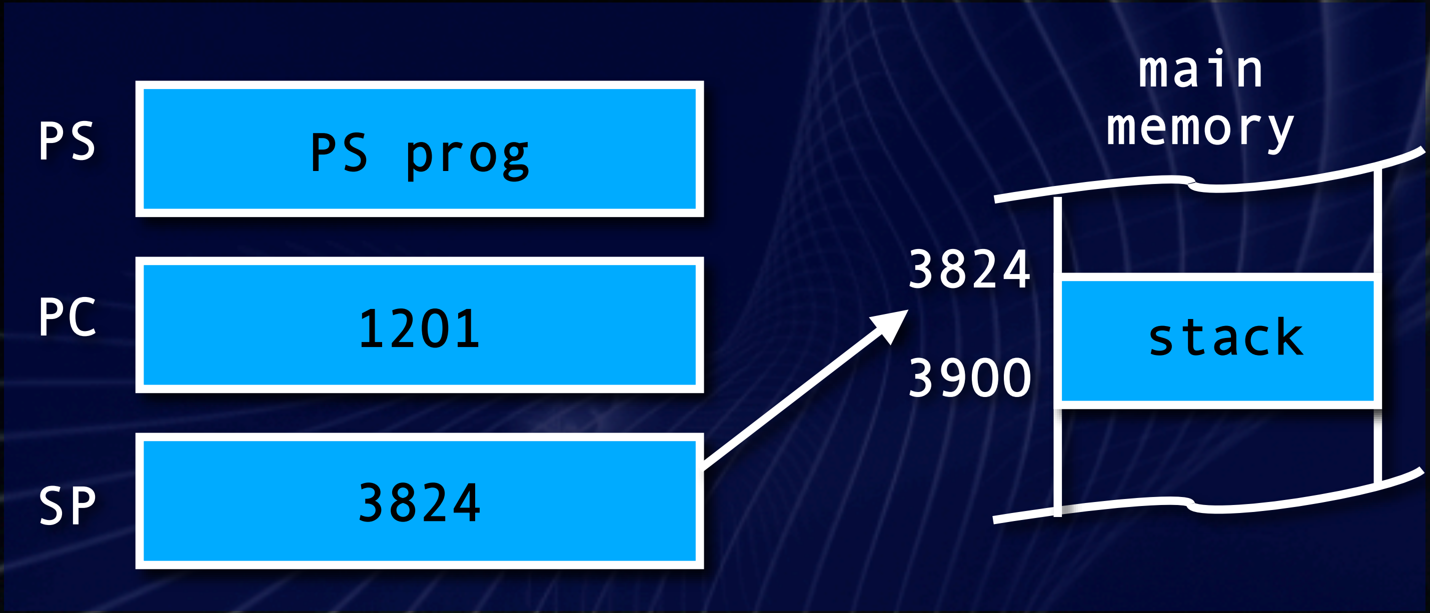
→STI (Set Interrupt)
 CLI (Clear Interrupt)

→Hardware priority

Interrupt processing

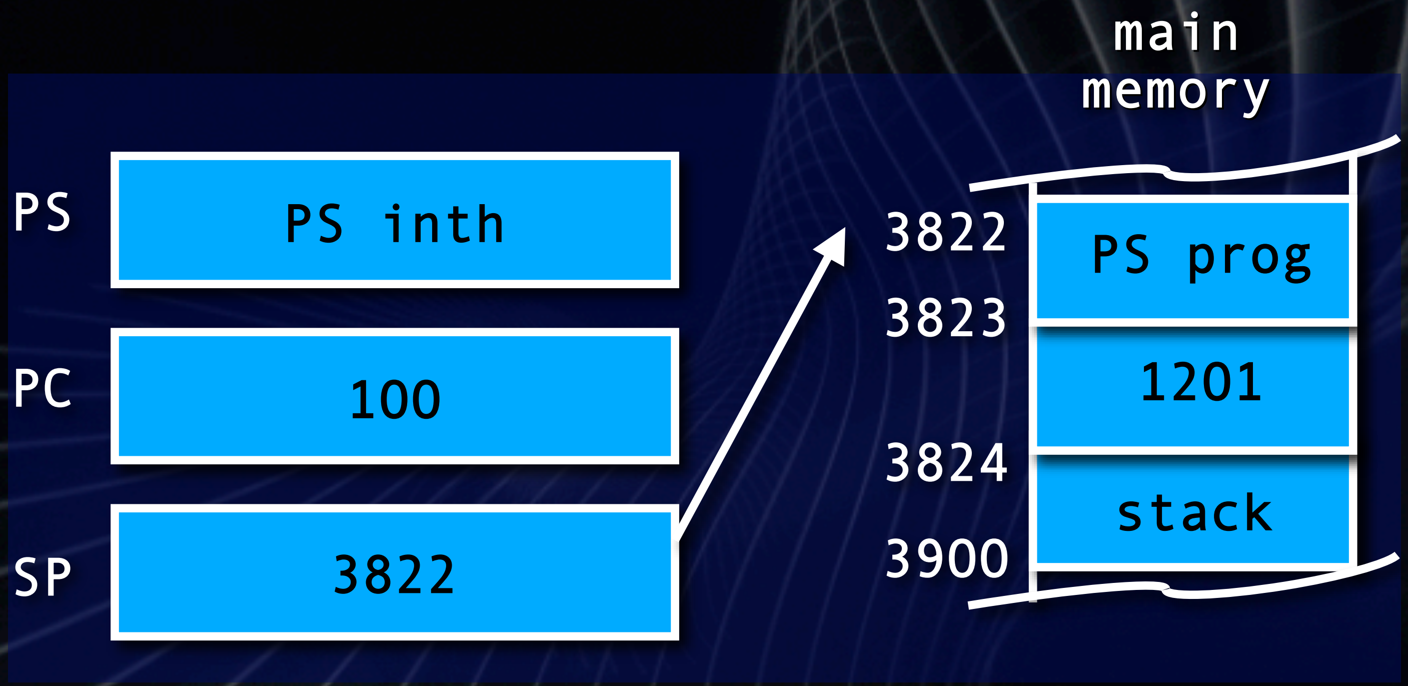


Interrupt processing



CPU completes execution of the current instruction

Interrupt processing



main memory

PS

PS inth

PC

100

SP

3822

3822

3823

3824

3900

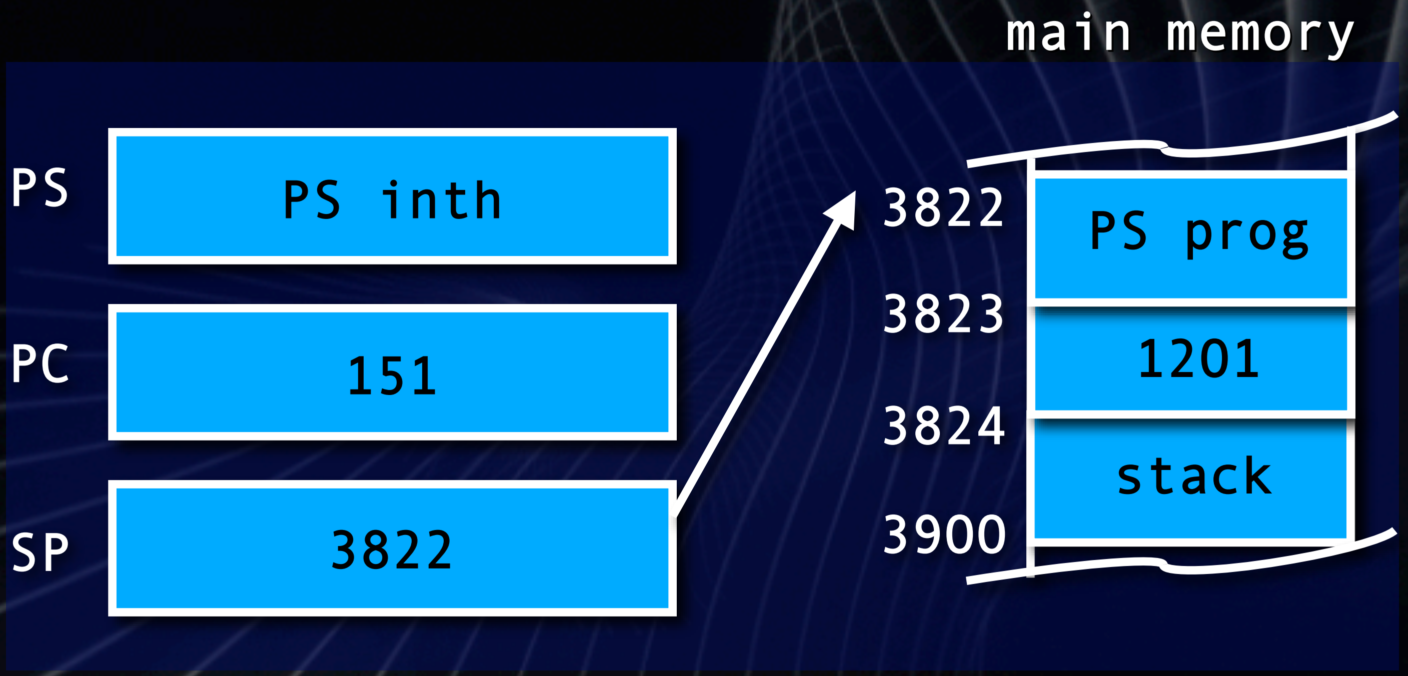
PS prog

1201

stack

interrupt signal has been accepted

Interrupt processing



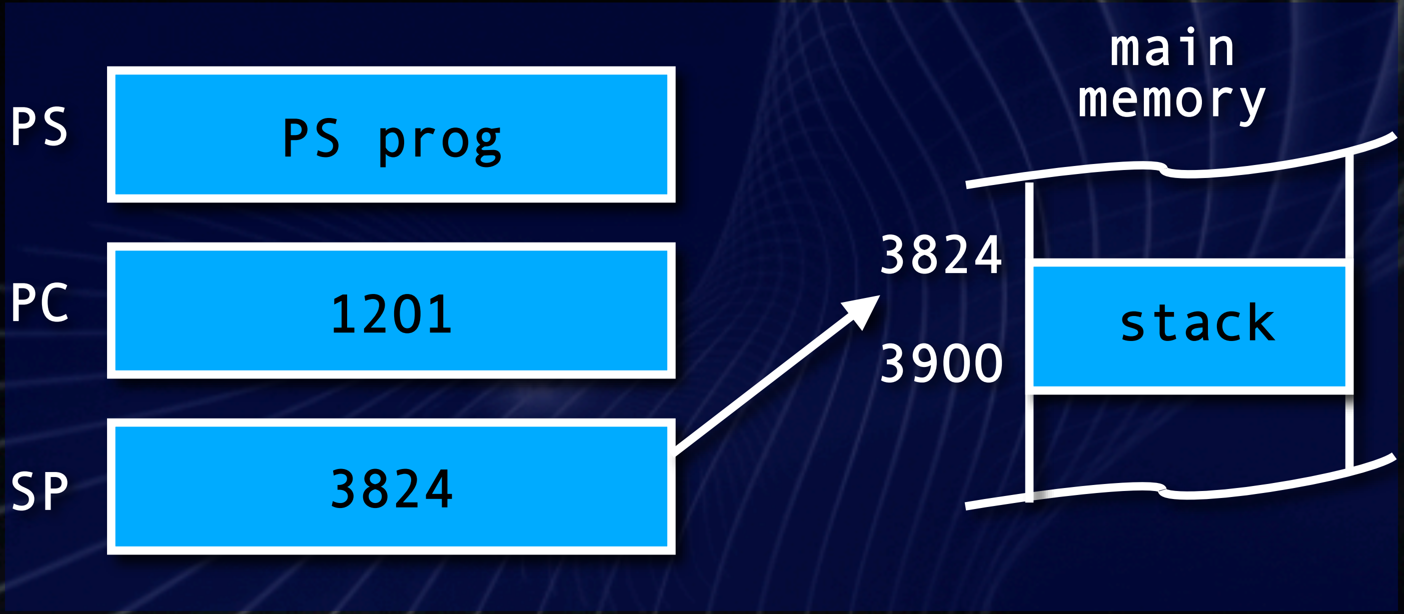
main memory

PS	PS inth
PC	151
SP	3822

3822	PS prog
3823	1201
3824	stack
3900	

iret execution

Interrupt processing



iret completed

Contents

O.S.Components

O.S. COMPONENTS

→ Process Management

→ Main-Memory Management

→ Secondary-Memory Management

O.S. COMPONENTS

→I/O-System Management

→File Management

Process management

→ Creating and deleting processes

→ Suspending and resuming processes

→ Provide mechanisms for process interaction

Main memory management

Keeping track of which parts of memory are currently being used and by whom

Main memory management

Deciding which processes are to be loaded when memory becomes available

Main memory management

Allocating and deallocating memory space as needed

Secondary memory management

→ Free space management

→ Storage allocation

→ Disk scheduling

I/O SYSTEM MANAGEMENT

The I/O subsystem consists of

→ A memory-management component that includes buffering and spooling

I/O SYSTEM MANAGEMENT

The I/O subsystem consists of

→ A general device driver interface

→ Drivers for specific hardware devices

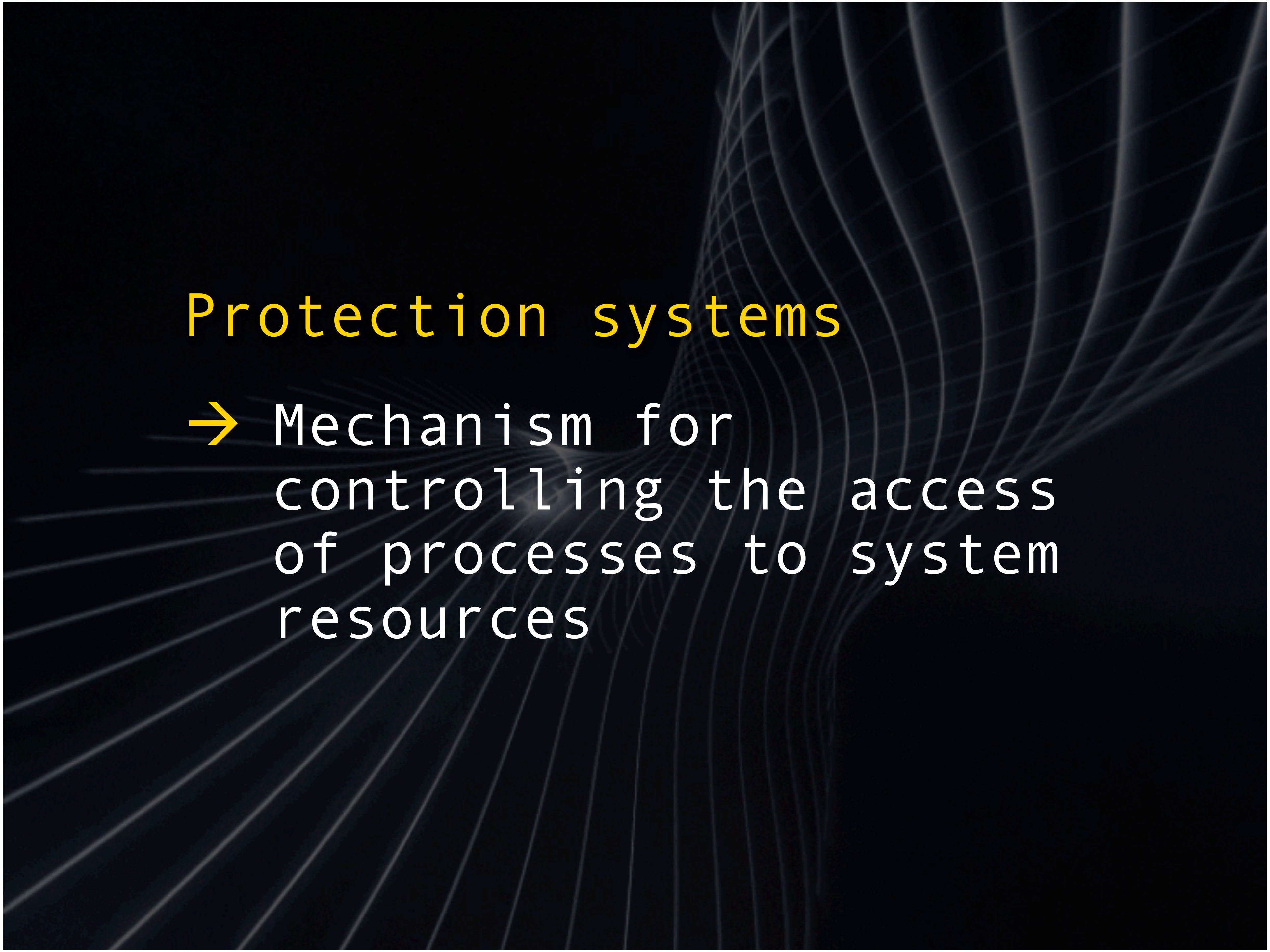
File management

→ Creating and deleting files and directories

→ Supporting primitives for manipulating file and directory

File management

- → Mapping files onto storage
- → Backing up files on stable storage media



Protection systems

→ Mechanism for controlling the access of processes to system resources

Protection systems

- → The mechanism must provide means for specifying the controls to be imposed and means for enforcement

Comand-Interpreter System

→ The interface between users and O.S.

→ A program that reads and interpretes control statements

Comand-Interpreter System

→ Its main function: to get next command statement and execute it (shell)

Contents

System calls

System Calls

Processes communicate with the O.S. and request services to it by making system calls

System Calls

System calls provide the interface between any process and O.S.

Examples of System Call

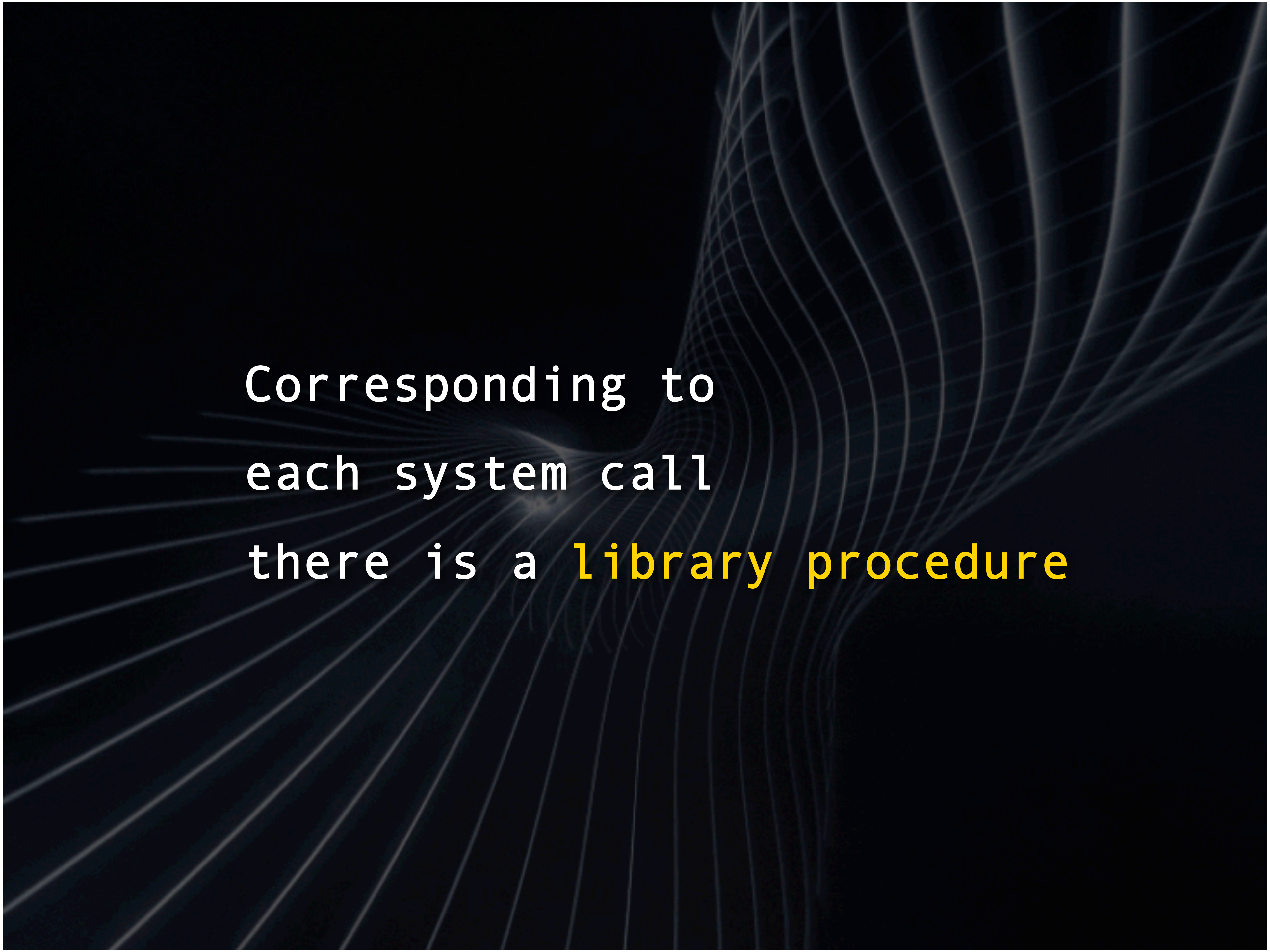
→Process control

→File manipulation

→Device manipulation

→Information management

→Communications



Corresponding to each system call there is a library procedure

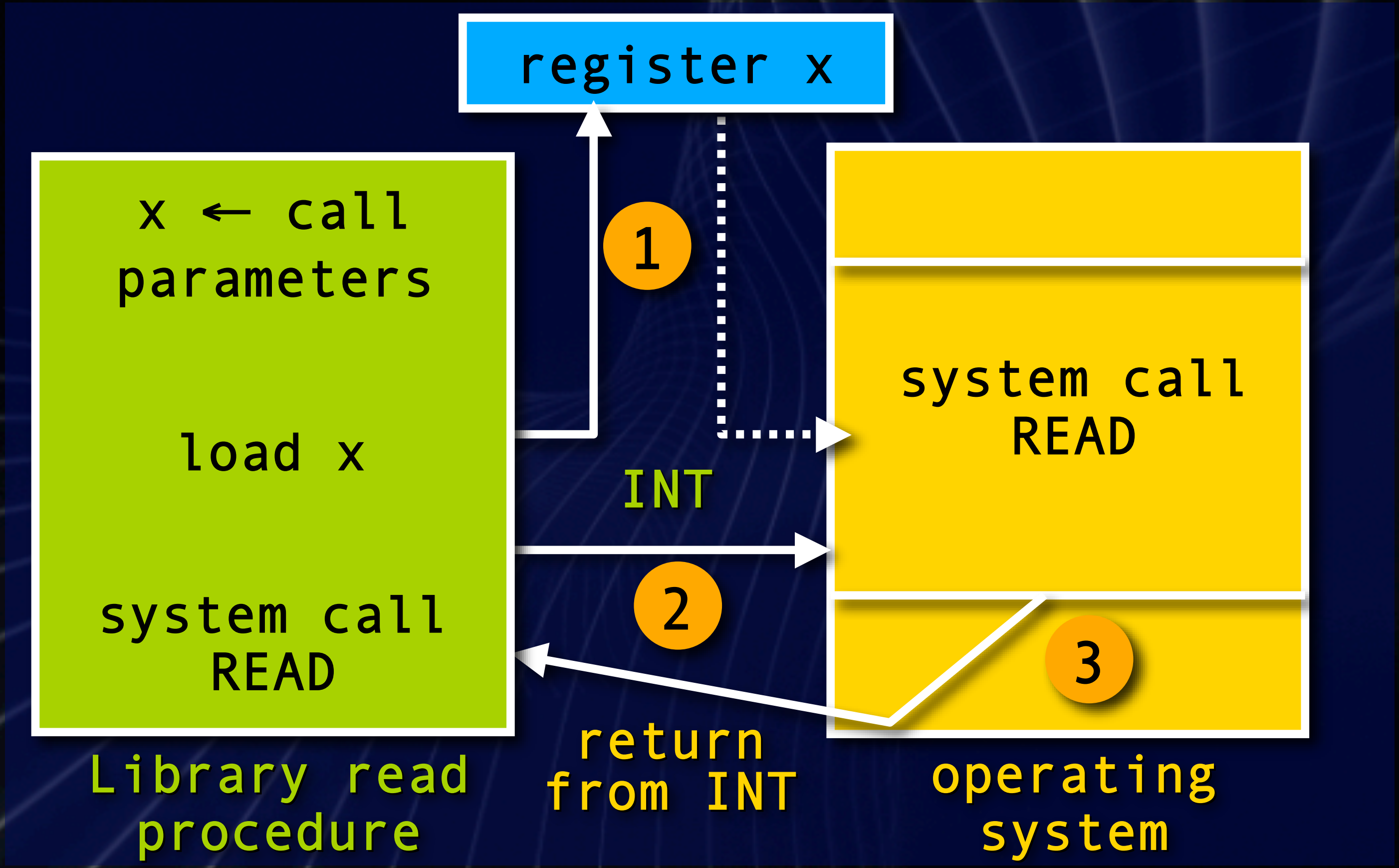
Example: read system call

C program

↓

```
count= read (file, buffer, nbytes)
```

count returns the number of bytes actually read

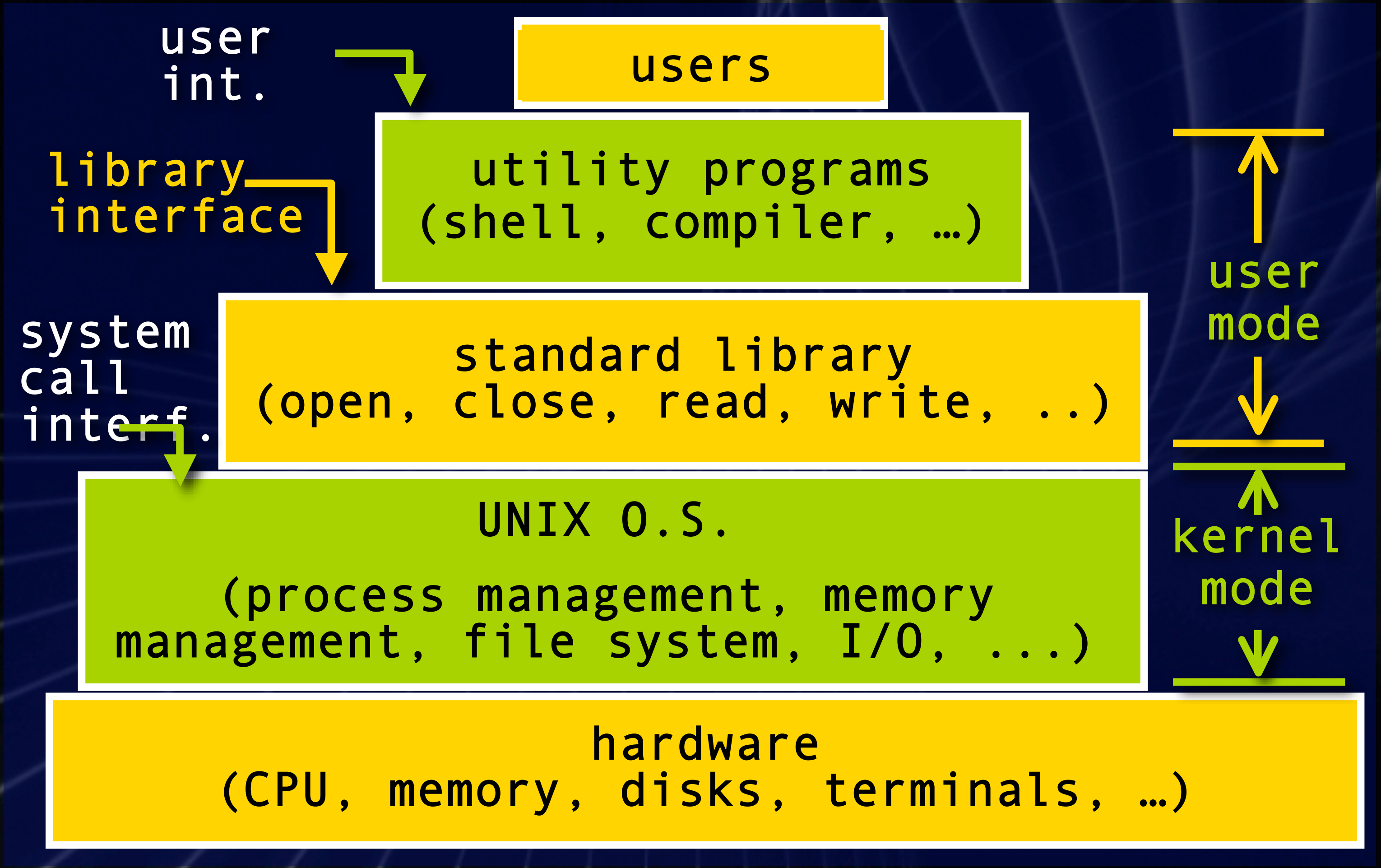

register x
x ← call parameters
load x
system call READ
Library read procedure
1
INT
2
return from INT
system call READ
3
operating system

The library procedure

- Puts the parameters of the system call in machine registers
- Issues a INT instruction to start the O.S.
- The control is returned to the caller by returning the status code as a result

Methods to pass parameters to operating system

- directly via registers
- in a memory block whose address is passed as a parameter in a registry
- into the stack (push, pop)



user int.

users

library interface

utility programs
(shell, compiler, …)

system call interf.

standard library
(open, close, read, write, ..)

UNIX O.S.

(process management, memory management, file system, I/O, ...)

hardware
(CPU, memory, disks, terminals, …)

user mode

kernel mode

Dual mode of operations

user mode

➔ Used for normal execution of user programs

➔ Hardware control does not allow the execution of privileged instructions

Dual mode of operations

supervisor mode (kernel mode)

- Used for the execution of O.S. functions as required by system calls

- All instructions can be executed