



# OPERAČNÍ SYSTÉMY II.

**P2**  
**2006-10-10**

## Operating systems consists of:

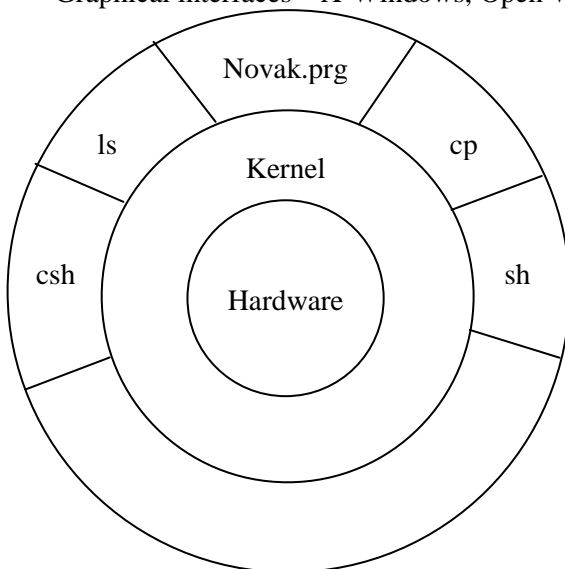
- ✓ Kernel
- ✓ System programs

## Kernel tasks:

- ✓ Creates processes and controls their execution.
- ✓ Facilitates communication among processes
- ✓ Provides means for work with I/O devices.
- ✓ Part of the kernel is a file manager, which organizes disc data into files and directories.
- ✓ Overviews the system and records logs and different statistics.

## Unix consists of:

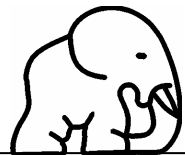
- ✓ Kernel
- ✓ System programs and libraries:
  - Shells – sh, bash, ksh, csh, tcsh.
  - Programs for file and directory manipulation – ls, cp, rm, tar etc.
  - Programming support – compilers, libraries – podpora programů.
  - Communication support – telnet, ftp, rlogin etc. – podpora komunikace v sítích.
  - Graphical interfaces – X-Windows, Open Windows – grafický interpret.



## Basic concepts:

- ✓ **Program:** executable file on disc, soubor na disku.  
Fig. structure of a target program on disc:

Magic number
Header
Text of program
Initialized data



- ✓ **Process:** running program.

Stack
Noninitialized data
Initialized data
Text

### OS activity after interrupt:

#### Interrupt:

- ✓ External: I/O devices
- ✓ Inner:
  - Processor running errors – i.e. overflow, dividing by zero
  - ?

**System services (calls)** – every OS provides services to processes. System services can be realized by a subroutine call or internal interrupt.

- ✓ System service in Unix are realized by means of internal interrupt.
- ✓ System service is a function that contains instruction for internal interrupt. As soon as this instruction is executed the process changes the mode of running: it passes from user to kernel running. In kernel mode process executes code of the kernel (code of the system call).
- ✓ System calls cannot be programmed in C language, they must be programmed in Assembler.

#### Preemption:

- ✓ Process can run continuously only for limited time interval (time quantum). After this interval is expired kernel stops the process and puts it ready to run state. Then scheduler chooses another process for running. The former process is said to be preempted.
- ✓ **The kernel is not preemptive:** If a process is running in kernel mode, it cannot be preempted. The reason is to exclude a possibility of time dependence of processes (race conditions). Process running in kernel mode at first finishes kernel running (system service) and when it returns to user running, the control is given to scheduler and a new process could be scheduled.
- ✓ **Context switch:** If another process is scheduled, kernel must swap user and system contexts of old process for user and system contexts of new one.

#### External interrupts:

- ✓ External interrupts are executed in the context of the interrupted process. Only system context of interrupted process must be saved and after return from interrupt it must be restored.

**Example (Unix)** – process A executes system service read() - reading from disc:

- ✓ Read() generates internal interrupt. Code of system calls starts to be executed.
- ✓ If asked data are not in disc cache in main memory, interrupt handler starts I/O transfer. Kernel puts process A into asleep state.
- ✓ Process B is scheduled and begins to run.
- ✓ Transfer from disc is finished. Disc module interrupts processor.
- ✓ Interrupt is allowed. Disc interrupt handler is executed. Execution of the handler code is done in context of process B. At the end of its running interrupt handler puts process A into ready to run state.
- ✓ Return from interrupt. Process B continues to run.
- ✓ Time quantum for process B has expired and B is preempted. Process A is scheduled.
- ✓ Běží proces A, potřebuje data z disku, skočí do OS, ten inicializuje čtení z disku, které je nezávislé na činnosti procesoru, proces A je zablokovaný a spouští se jiné procesy.



### Editor vi:

- ✓ :w – zapsat
- ✓ :g – opustit
- ✓ :wg
- ✓ :w a.c
- ✓ :g! – opustit bez zápisu
- ✓ :set number – očíslování řádek
- ✓ :set nonumber – zrušení číslování řádek
- ✓ 2dd – zrušit
- ✓ yy – přepsat do vyrovnávací paměti
- ✓ p – vypsání za kurzor
- ✓ P – vypsání před kurzor
- ✓ h, j, k, l – posun kurzoru

