

# Program předmětu X36 DIS

- **Obecný úvod:**
  - Základní pojmy a souvislosti, typy modelů a jejich vlastnosti, práce s časem.
- **Modelování systémů hromadné obsluhy**
  - Charakteristika systémů hromadné obsluhy.
  - Obecné principy simulačních systémů diskrétního typu.
  - Nástroje:
    - výukový systém v C++,
    - simulační systém OMNET++.
  - Metoda Monte Carlo, generování, transformace a testování pseudonáhodných čísel,
  - Paralelní simulace: konservativní a optimistické metody,
  - Analytické modely systémů hromadné obsluhy.

# Literatura

1. Jerry Banks, John S. Carson, Barry L. Nelson, David M. Nicol : Discrete-Event System Simulation, Prentice Hall 1999, 2001, 2005.
2. George S. Fishman: Discrete-Event Simulation: Modeling, Programming and Analysis, Springer-Verlag, Berlin 2001.
3. Richard M. Fujimoto: Parallel and Distributed Simulation Systems, John Wiley & Sons, Inc. N.York, 2000.
4. Donald E. Knuth: The art of Computer Programming, Volume 2, third edition, Addison –Wesley 1998.
5. Leonard Kleinrock: Queuing systems, Volume 2, Computer Applications, John Wiley & Sons, Inc. N.York, 1995.
6. Maik Schwarz: "Telecommunication Networks, Protocols, Modeling and Analysis," Addison-Wesley, 1987..

# Osnova přednášek

- 1) Základní pojmy a souvislosti. Úvod do modelování a simulací.
- 2) Systémy hromadné obsluhy: obecná charakteristika. Principy simulačních systémů diskrétního typu.
- 3) Koncepce výukového simulačního systému a principy kvaziparalelního prostředí.
- 4) Příklady aplikace procesově orientovaného kvaziparalelního prostředí.
- 5) Možnosti rozšíření výukového systému pro speciální účely.
- 6) Systém OMNET++. Popis struktur v jazyku NED.
- 7) Specifikace funkčních modulů. Komunikační funkce systému OMNET++.
- 8) Koncepce zpráv, front, shromažďování statistik.
- 9) Aplikace systému OMNET++.
- 10) Metody generování pseudonáhodných čísel.
- 11) Transformační metody, testování.
- 12) Paralelní simulace: Konzervativní metody.
- 13) Paralelní simulace: Optimistické metody.
- 14) Analytické modely jednoduchých systémů hromadné obsluhy.

# Osnova cvičení

- 1) Úvod do simulací. Základní principy.
- 2) – 3) SHO: příklady na dekompozici systémů:
  - orientace na události a procesy,
  - vývojové diagramy, synchronizace
  - zadání domácího úkolu.
- 4) – 6) Řešení úloh z oblasti systémů hromadné obsluhy:  
( nástroj: výukový simulační systém ).
- 7) Odevzdání a kontrola 1. projektu.
- 8) Úvod do systému OMNET++, seznámení s prostředím.
- 9) Popis rozhraní a propojení modulů, zadání 2. projektu.
- 10) – 13) Řešení příkladů v systému OMNET.
- 14) Odevzdání 2. projektu, zápočet.

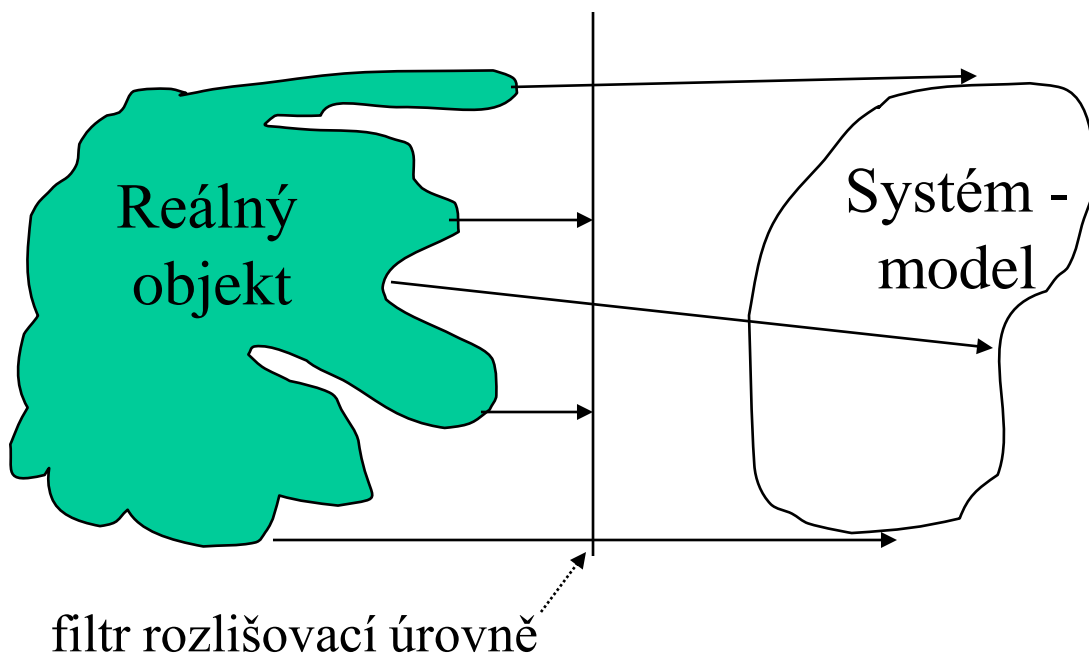
# Základní pojmy

Zkoumání světa: nutné třídění, rozlišování, zjednodušování, izolace

**Objekt ( Entita )** - část okolního světa

**System:**

- pojem intuitivně užívaný v různých souvislostech - charakteristika:
  - účelově definovaný na objektu (záměr)
  - stanovení rozlišení úrovně,
  - vymezení části objektu a stanovení vnitřní organizace objektu,
  - odráží naše vidění reálného objektu.
- matematicky specifikovaný termín v teorii systémů a modelování
- příklady:
  - doprav. systém Prahy: tram., autob, metro,..
  - zaživ. systém člověka: žalud., střeva, játra,..
  - paměť počítače: registry, cache, oper. pam,..



# Základní pojmy

**Modelování** - stručná charakteristika:

- obecné metoda společná všem vědám
- tvorba systému - modelu na reálném objektu
- cílevědomá činnost sloužící k získávání poznatků o reálném objektu

**Model:**

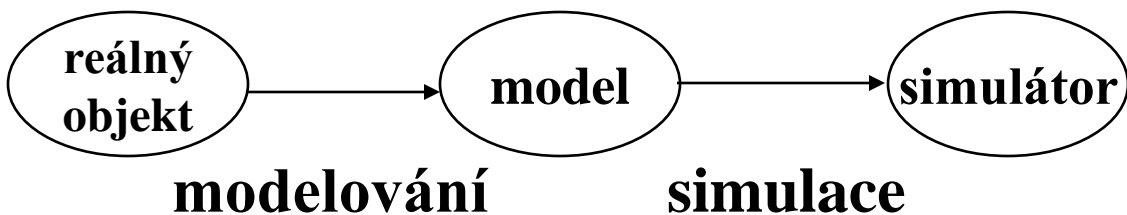
- nejširší pojetí: metodologický nástroj (velmi volně řečeno); př: dílenský výkres, schéma obvodu, mapa terénu, atd.
- užší pojetí: zaměříme se pouze na matematické modely :
  - striktně matematicky specifikovaný systém na určité úrovni abstrakce (viz 2. část)
    - analytické modely (př.: Ohmův zákon)
    - numerické modely (simulační modely)
  - neuvažujeme modely založené na fyzikální podobnosti

**Simulace**

- numerické řešení matematických modelů

**Simulátor (simulační systém)**

- výpočetní systém (počítač, člověk,..) provádějící numerický výpočet chování modelu



# Základní pojmy: formální specifikace systému

Formalismus DEVS (discrete event system specification)

$DEVS = ( X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta )$

$X$ .....množina vstupních hodnot,

$Y$ .....množina výstupních hodnot,

$S$ .....množina stavů  $s$ ,

$Q = \{ (s, e), s \in S, 0 \leq e \leq ta(s) \}$ ...množina celk. stavů  $q = (s, e)$ ;

$e$  ...doba strávená ve stavu  $s$  (měřená od okamžiku přechodu do daného stavu),

$\delta_{ext} = Q \times X \rightarrow S$ ...externí stavově přechodová funkce: definuje přechody v závislosti na vnějších vstupech,

$\delta_{int} = S \rightarrow S$ .....vnitřní stavová funkce: definuje přechody v závislosti na vnitřní funkci času,

$\lambda = S \rightarrow Y$ .....výstupní funkce,

Poznámky:

*bezprostředně po přechodu do stavu  $s$ , je celkový stav  $q = (s, 0)$ .*

- nastane-li v čase  $e$  ( $0 \leq e \leq ta(s)$ ), měřeno od vstupu do stavu  $s$ ) vnější událost, pak nastane přechod dle externí přechodové funkce,*
- nenastane-li v čase  $e$  ( $0 \leq e \leq ta(s)$ ), měřeno od vstupu do stavu  $s$ ) vnější událost, pak v čase  $e = ta(s)$  nastane přechod dle vnitřní přechodové funkce,*
- hodnotu funkce  $ta(s)$  lze definovat v každém stavu  $s$ :  
 $ta(s) = S \rightarrow R_0^+$  ,  $R_0^+$  .....množina nezáporných čísel*

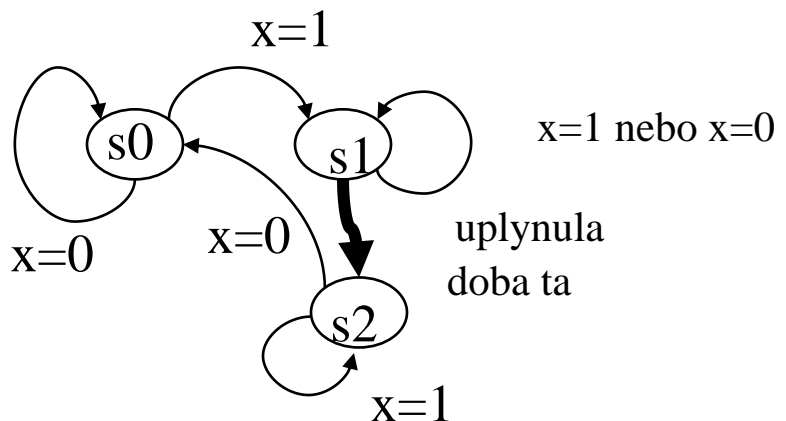
# Příklad formální specifikace systému

## Formální definice DEVS:

- definuje funkci systému nebo jeho části - umožňuje predikovat výstup na základě znalosti vstupu,
- pro strukturní modely je třeba takto definovat funkci všech dílčích prvků systému a vzájemné relace mezi nimi ( propojení + interakce )

## Příklad - monostabilní klopný obvod:

- náběžná hrana vstupu  $x$  spouští výstupní puls šířky 2.7ns,
- během výstupního pulsu klopný obvod ignoruje všechny startovací hrany vstupního signálu  $x$



### **begin**

s0: výstup = 0;

čekej na náběžnou hranu  $x$ ;

nastav  $ta(s1) = 2.7 \text{ ns}$ ; // doba setrvání ve stavu s1

s1: výstup = 1;

čekej po dobu  $ta$ ;

nastav  $ta(s2) = \infty$ ; // možná doba setrvání ve stavu s2

s2: výstup = 0;

if  $x=1$  čekej na závěrnou hranu  $x$ ;

nastav  $ta(s0) = \infty$ ; // možná doba setrvání ve stavu s0

goto s0;

### **end;**



# Základní pojmy

- prvek systému - dále nedělitelná část systému (modelu)
- atributy systému - veličiny charakterizující prvky systému (modelu)
- chování systému - průběh hodnot atributů v čase či prostoru
- okolí systému - zdroj podnětů na systém (model)
- interakce - působení mezi prvky systému (modelu) navzájem a mezi prvky a okolím

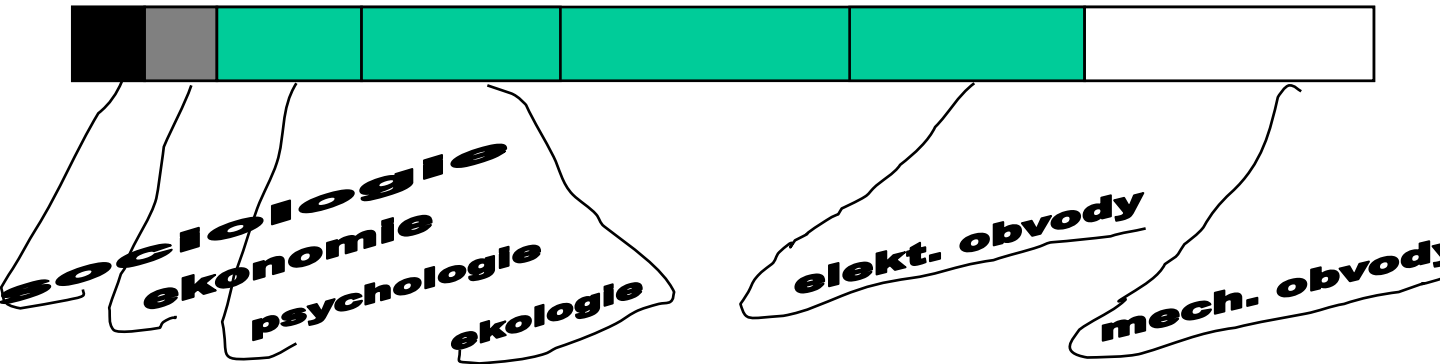
## Třídění systémů (modelů): dle nejrůznějších hledisek

- dle podobnosti:
  - fyzikální (malá přehrada, čelní náraz auta)
  - **matematické**
    - **analytické**
      - nositel teorie
      - ucelený pohled na reálný objekt
      - př.: model vesmíru, Ohmův zákon
    - **simulační** - numerické
      - jde o metodu řešení problémů
      - lze častěji aplikovat (nástroj hrubé síly)
      - funkční i strukturní modely

# Vztah modelu a originálu

So....originál, Sm....model

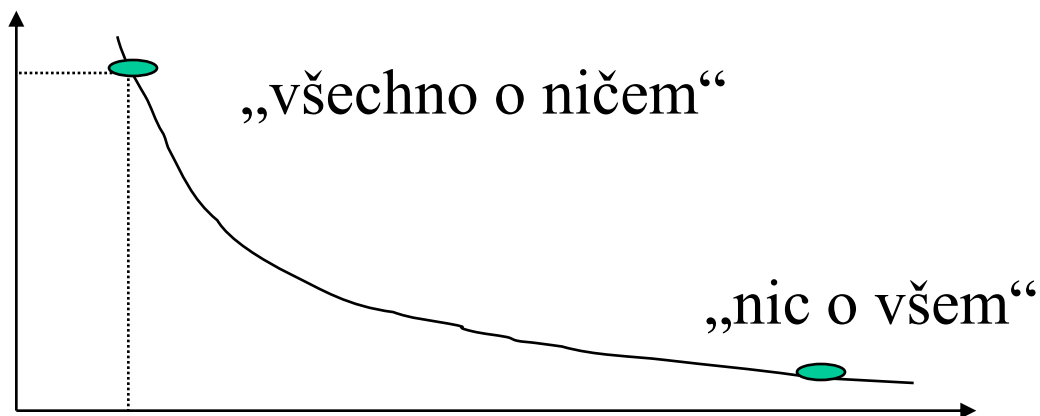
- funkční (funkcionální) model (model chování)
  - So, Sm.....stejné chování
- strukturní model
  - So, Sm....stejná struktura a stejné chování



Obecně návrh modelu: závisí na účelu, často i na úrovni poznání

- velké zjednodušení - ztrácí podstatné rysy
- velké podrobnosti - nezvládnutí úroveň podrobností

úroveň podrobností



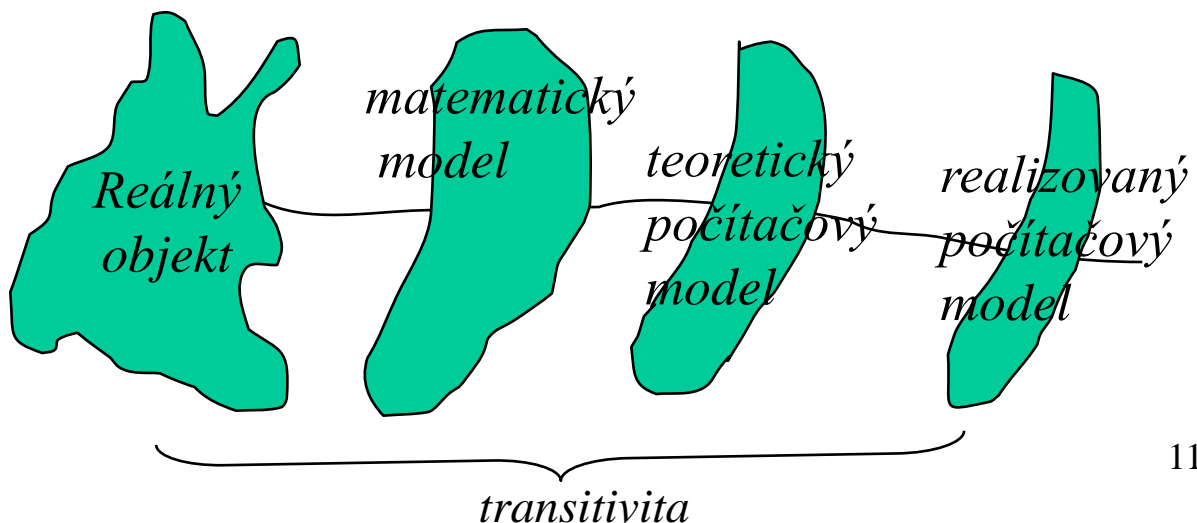
Složitost simulovaného objektu

# Rozdělení modelů

## Další hlediska třídění:

- dle interakce s okolím:
  - otevřené (interakce s okolím)
  - uzavřené (bez interakce s okolím)
- dle vnitřní paměti:
  - statické - (bez vnitřní paměti; př. kombinační obvod)
  - dynamické - (s vnitřní pamětí; př. sekvenční obvod)
- dle chování:
  - spojité
  - diskrétní
  - kombinované
  - stochastické
- zavedené formalismy systémů (př.: DESS, DTSS, DEVS, DEV&DESS,..)

## Transitivita modelů:



# Funkce modelů

- nástroj myšlení (př.: konečný automat - minimalizace vnitřních stavů)
- prostředek ke komunikaci : odborné konference, časopisy
- nástroj výuky : trenažery,
- prostředek předvídání a experimentování:
  - analýza .....chování ?
  - řízení .....vhodné působení ?
  - identifikace ...parametry ?
  - syntéza.....struktura ?

## Etapy konstrukce modelů:

1. formování účelového popisu
  - obecně nejobtížnější krok - neexistuje návod ,definice systému na reálném objektu, účel modelu, specifikace prvků, parametrů a funkčních vztahů,
  - poměrně snadné v technických vědách
2. realizace modelu na počítači: použití podpůrných nástrojů
3. provádění simulačního programu („simulace v užším smyslu“)

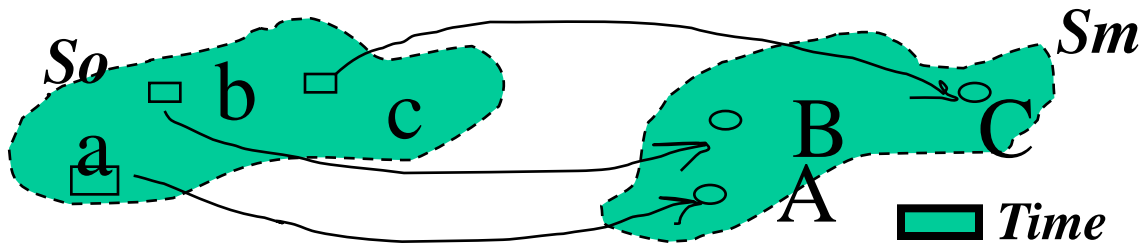
Poznámka: realita - prolínání výše uvedených etap

# Simulační systémy – obecná charakteristika

- pracovní nástroje - nabízí možnosti pro snadný návrh simulačních programů:
- dekompozice výsledného simulačního algoritmu na dílčí samostatné části (popisující logicky paralelní procesy okolního světa) => přirozené paralelní vidění simulovaného systému,
- automatické udržování modelového času,
- automatizace návazností dílčích procesů:
  - zajišťuje synchronizační jádro,
  - závisí na celkové koncepci nástroje,
  - kvaziparalelní prostředí ( jeden sdílený procesor ):
    - snadné plánování časově závislých procesů,
    - použitá strategie: vždy **kooperativní** sdílení procesoru
    - další oblasti kvaziparalelního prostředí:
      - multi-uživatelské operační systémy,
      - operační systémy reálného času
  - paralelní prostředí (několik sdílených procesorů):
    - „složitá synchronizace časově závislých procesů,  
důvod: distribuovaný plán časově závislých procesů  
( viz. později: paralelní simulace )

# Časová synchronizace procesů

- jde o synchronizaci vzhledem k modelovému času: hodnoty reálného času mapujeme na hodnoty z množiny čísel (real nebo integer) a okamžitou hodnotu modelového času uchováváme v proměnné Time,



$S_o$ : originál: atributy:  $a, b, \dots$ , reálný čas:  $t$

$S_m$ : model: proměnné:  $A, B, \dots$ , modelový čas: Time

- proměnná Time nabývá postupně neklesajících hodnot  $T_j$ ,
- proměnné  $A, B, \dots$  modelu  $S_m$  representují hodnoty atributů  $a, b, \dots$  originálu  $S_o$  v čase  $t_j$  pokud platí  $Time = T_j$

## Přístupy k plánování procesů:

- **implicitní** (např. VHDL, VERILOG): automaticky vyplývá z popisu struktury simulovaného systému,
- **explicitní** (např. náš výukový systém):
  - neexistence popisu struktury,
- **kombinace předchozích**: OMNET++

## Vzájemná komunikace procesů:

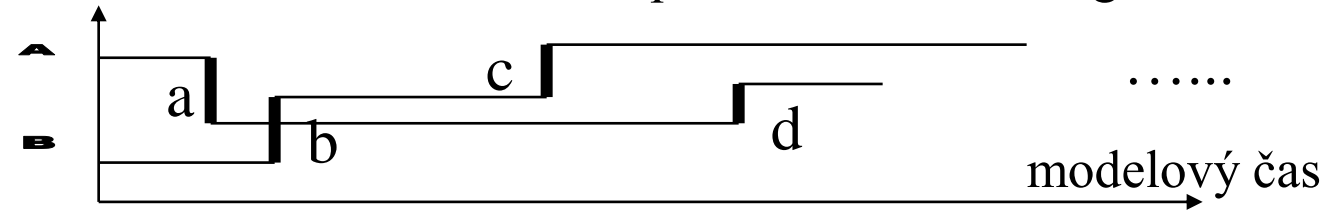
- **sdílení dat (data sharing)**: př.: náš výukový systém v C++
- **předávání zpráv (message passing)**: OMNET++

# Rozlišení časů

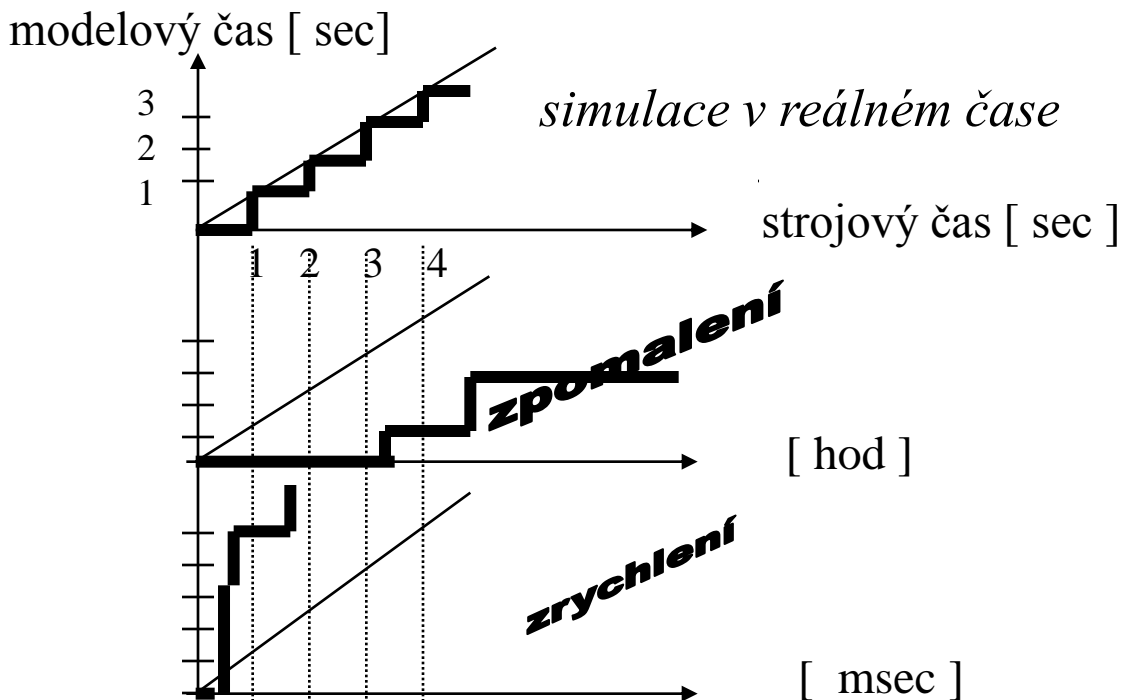
- reálný čas: čas, vůči němuž vyšetřujeme chování simulovaného objektu,
- strojový čas: reálný čas spotřebovaný při simulaci,
- simulační (modelový čas): čas vůči němuž posuzujeme chování simulačního modelu (obraz reálného času v modelu)

## Časová synchronizace procesů:

- dle modelového času : kooperativní multitasking



## Vztah modelového a strojového času



# Simulační systémy - rozdělení

## Simulační systémy:

- bez podpory popisu struktury: obecné kvaziparalelní prostředí s orientací na :
  - události: SIMSCRIPT, ..... ..
  - sledování aktivit CSL, .
  - diskrétní procesy: GPSS, SIMULA I, SIMULA 67, DEMOS, SIMC,.....
  - diskrétní události a spojité procesy (kombinované systémy):  
GASP, GSL,.....
- s podporou strukturního popisu:
  - specializované na :
    - sítě: OMNET ++, NS-2, ....
    - číslicové obvody: OrCAD, VHDL, Verilog, SystemC, SystemVerilog,....
    - spojité systémy: CSMP, ACSL,....
    - analogové obvody: SPICE,.....
    - hybridní elektrické obvody: ANALOG ARTIST, VHDL-AMS,.....

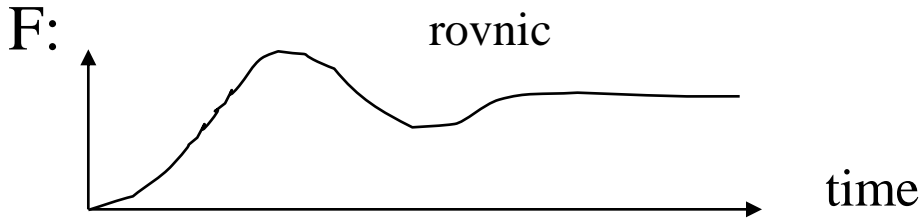


# Diskrétní a spojité procesy

- Událost (event) - okamžitá změna stavu systému ( z hlediska modelového času)
- Proces (process) - posloupnost několika událostí v závislosti na modelovém čase

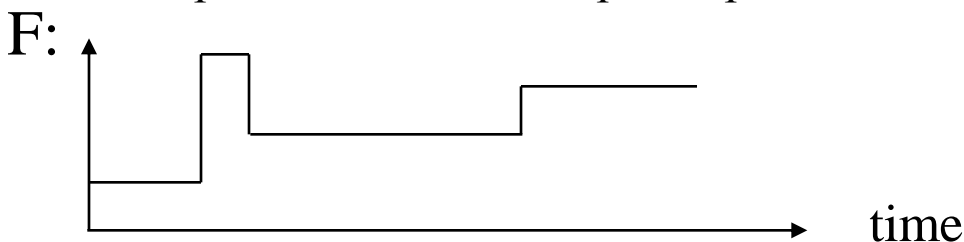
popis procesu - zobrazení  $F$  z časové množiny do množiny atributů systému

- spojité proces:  $F$ .....formou soustavy diferenciálních rovnic



- použitá metoda simulace: pevný časový krok ( analogie filmové kamery )

- diskrétní proces:  $F$ .....formou posloupnosti událostí



- používaná metoda simulace: proměnný časový krok + plánování událostí