

VNORENÉ POČÍTAČOVÉ SYSTÉMY

- **NEODDELITELNÁ SÚČASŤ RIADENÝCH M+E ZARIADENÍ**
- **ŠIROKÁ OBLASŤ POUŽITIA**
- **ZÁSADNÉ ZJEDNODUŠENIE M+E KONŠTRUKCIE**
- **INTELIGENTNÉ ZARIADENIA (NOVÁ FUNKCIONALITA, AmI, IoT)**

AMBIENT INTELLIGENCE

INTELIGENTNÉ PROSTREDIE

- VZÁJOMNE KOMUNIKUJÚCE
INTELIGENTNÉ ZARIADENIA
- ROZPOZNAJÚ PRÍTOMNOSŤ
ČLOVEKA
- PRISPOSOBIA PROSTREDIE
JEHO POTREBÁM

AMBIENT INTELLIGENCE

All this implies that the near future will bring us kitcheners that automatically switch off hot plates when we leave the apartment or, even more astonishing, TVs that automatically switch to another TV channel if it “judges” by our facial expressions that we don’t like the show which is on at the moment.

Aml

Ambient Intelligence is based on three key technologies:

- Ubiquitous Computing,
- Ubiquitous Communication &
- Intelligent User Interfaces

Ubiquitous Computing

Ubiquitous Computing means the integration of microprocessors into everyday objects like furniture, clothes or toys.

Using ubiquitous computing, people will not be aware of the presence of computers, they will be in the background.

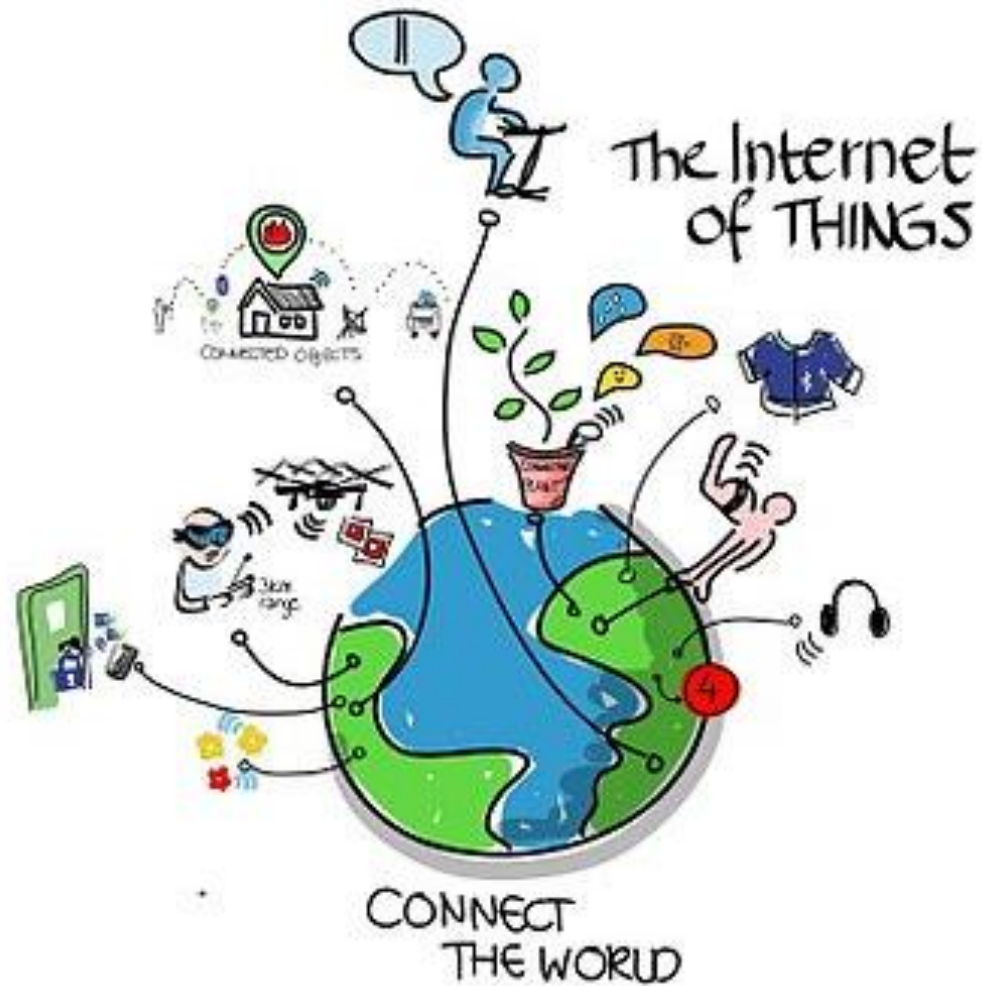
Ubiquitous Communication

Ubiquitous Communication should enable these objects to communicate with each other and with the user.

Intelligent User Interface

Intelligent User Interface enables the inhabitants of the Aml to control and interact with the environment in a natural (voice, gestures) and personalized way (preferences, context).

Internet of Things (IoT)



Internet of Things (IoT)

**PREPOJENIE
JEDNOZNAČNE
IDENTIFIKOVATEĽNÝCH ES
PROSTREDNÍCTVOM
INTERNETU**

Internet of Things (IoT)

- **R. 2020 – VIAC AKO 30 MLD.
ES PRIPOJENÝCH
K INTERNETU**
- **IPv6**

VNORENÉ POČÍTAČOVÉ SYSTÉMY

- **REAKTÍVNE SYSTÉMY**
- **PRÁCA V REÁLNO M ČASE**
- **ŠPECIALIZOVANÉ V/V ZARIADENIA**
- **BEZ OS, ŠPECIALIZ. SW V ROM (FIRMVÉR)**
- **MINIMÁLNE ROZMERY A SPOTREBA ENERIE**
- **ZVÝŠENÁ SPOĽAHLIVOSŤ**
- **ARCHITEKTÚRA**

VNORENÉ POČÍTAČOVÉ SYSTÉMY

PROCESORY PRE VS:

- **PROCESORY PRE PC**
- **OSTATNÉ PROCESORY**
- **JEDNOČIPOVÉ MIKROPOČÍTAČE**
- **SoC**

PROGRAMOVATELNÝ HW

- **SOFT-CORE PROCESORY**
- **JEDNOÚČELOVÉ SYSTÉMY**

VNORENÉ SYSTÉMY PRACUJÚCE V REÁLNO M ČASE

- **DOBA ODOZVY**
- **DEADLINE (MAX. DO)**
- **SYSTÉM REÁLNEHO ČASU**
- **ZLYHANIE SYSTÉMU**
- **SOFT RT SYSTÉMY**
- **HARD RT SYSTÉMY**
- **OBSLUHA POŽIADAVIEK - SÚBEŽNÉ
PROCESY**

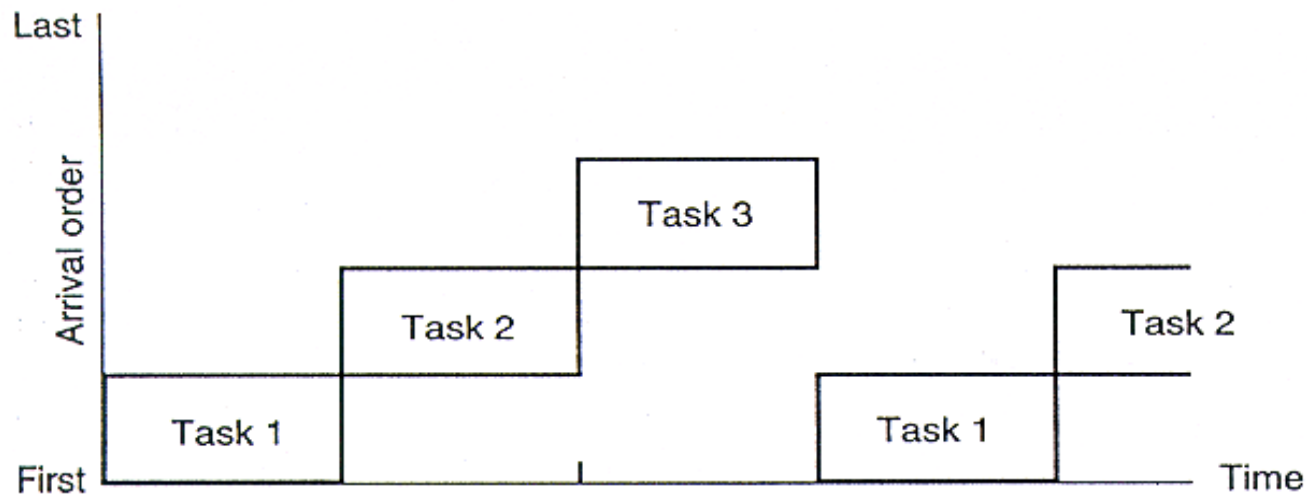
SÚBEŽNÉ PROCESY

- **IMPLEMENTÁCIA**
 - **PREPÍNANIE PROCESOV**
 - **PLÁNOVANIE PROCESOV**
- **NEZÁVISLÉ PROCESY**
- **SPOLUPRACUJÚCE PROCESY**
 - **KOMUNIKÁCIA**
 - **SYNCHRONIZÁCIA**

MECHANIZMY PREPÍNANIA PROCESOV

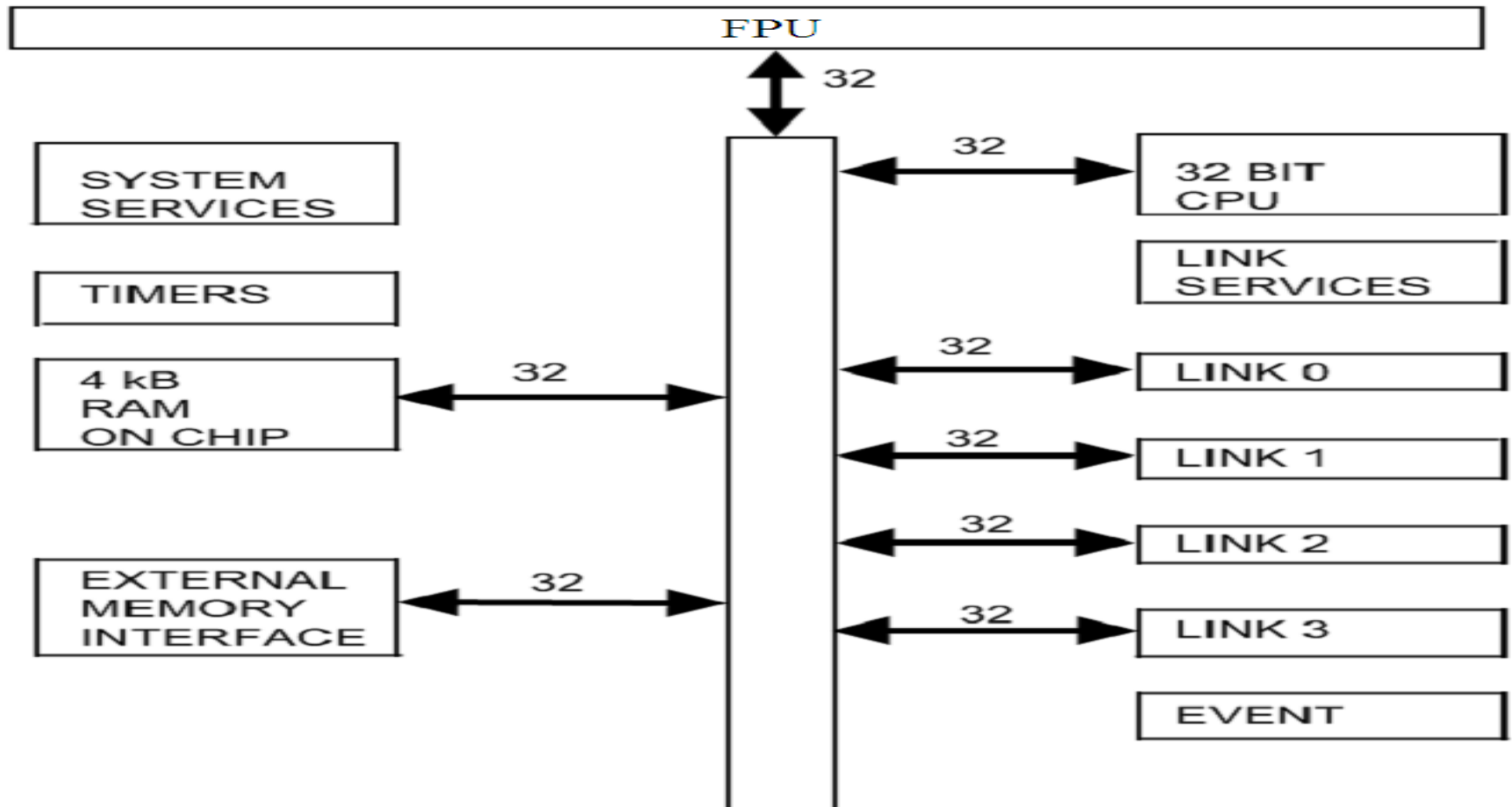
- **PRERUŠENIE:**
 - **PRIJATIE POŽIADAVKY**
 - **IDENTIFIKÁCIA ZDROJA**
 - **ODLOŽENIE STAVU BEŽIACEHO PROCESU**
 - **VYKONANIE PROCESU OBSLUHY PRERUŠENIA**
 - **OBNOVENIE STAVU PRERUŠENÉHO PROCESU**
 - **POKRAČOVANIE PRERUŠENÉHO PROCESU**
 - **x86 REAL MODE**

METÓDY PLÁNOVANIA PROCESOV



- **POSTUPNÉ PLÁNOVANIE**

HW IMPLEMENTÁCIA: TRANSPUTER



HW IMPLEMENTÁCIA: TRANSPUTER

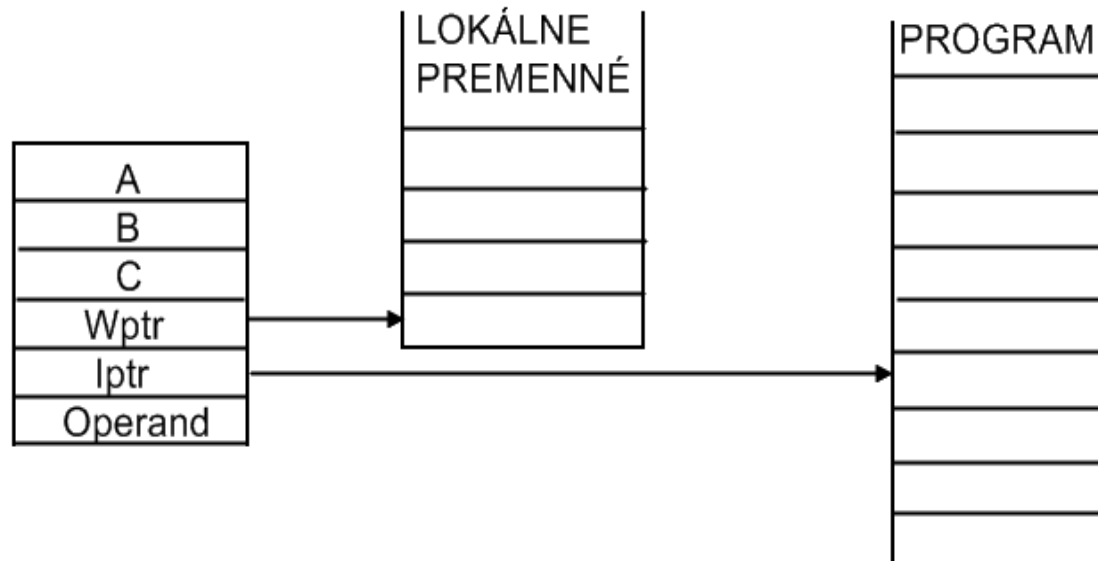
- **PRIORITNÉ ÚROVNE PROCESOV : (0, 1)**
- **ČASOVAČ 0: $1\mu\text{s}$ ($\approx 4295 \text{ s}$)**
- **ČASOVAČ 1: $64\mu\text{s}$ ($\approx 76 \text{ hod}$)**
- **KOMUNIKÁCIA MEDZI PROCESMI =
KANÁLY**
- **EVENT**
- **DESCHEDULING POINTS**

HW IMPLEMENTÁCIA: TRANSPUTER

- **DESCHEDELING POINTS:**
 - **INPUT MESSAGE**
 - **OUTPUT MESSAGE**
 - **JUMP**
 - **LOOP END**
 - **TIMER ALT WAIT, TIMER INPUT**
 - **END PROCESS, STOP PROCESS**
 - **STOP ON ERROR**
- **DESKRIPTOR PROCESU (PID)**

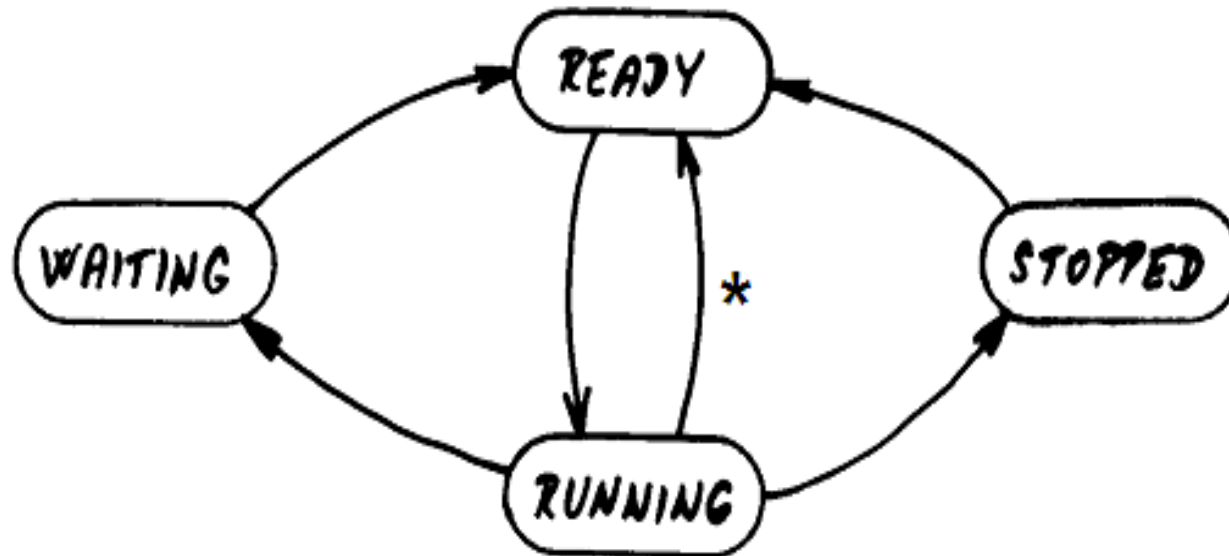
HW IMPLEMENTÁCIA: TRANSPUTER

- **REGISTRE PROCESORA:**



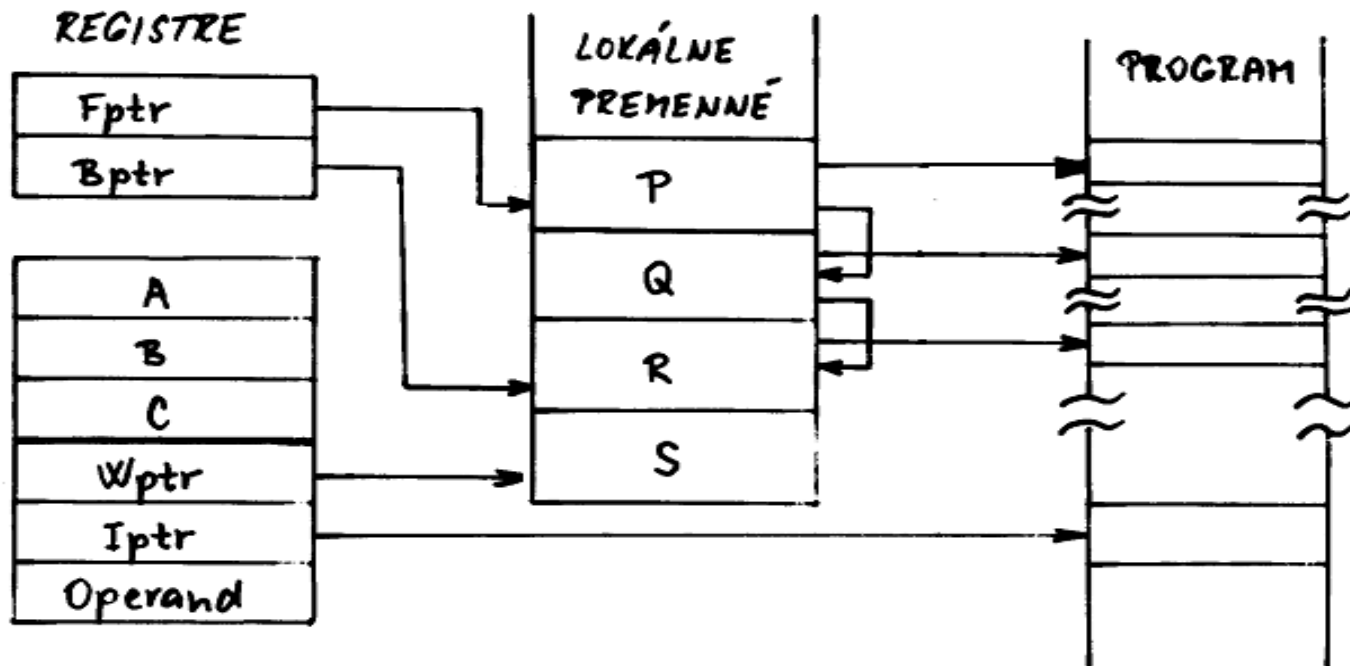
HW IMPLEMENTÁCIA: TRANSPUTER

- STAVOVÝ DIAGRAM PROCESOV



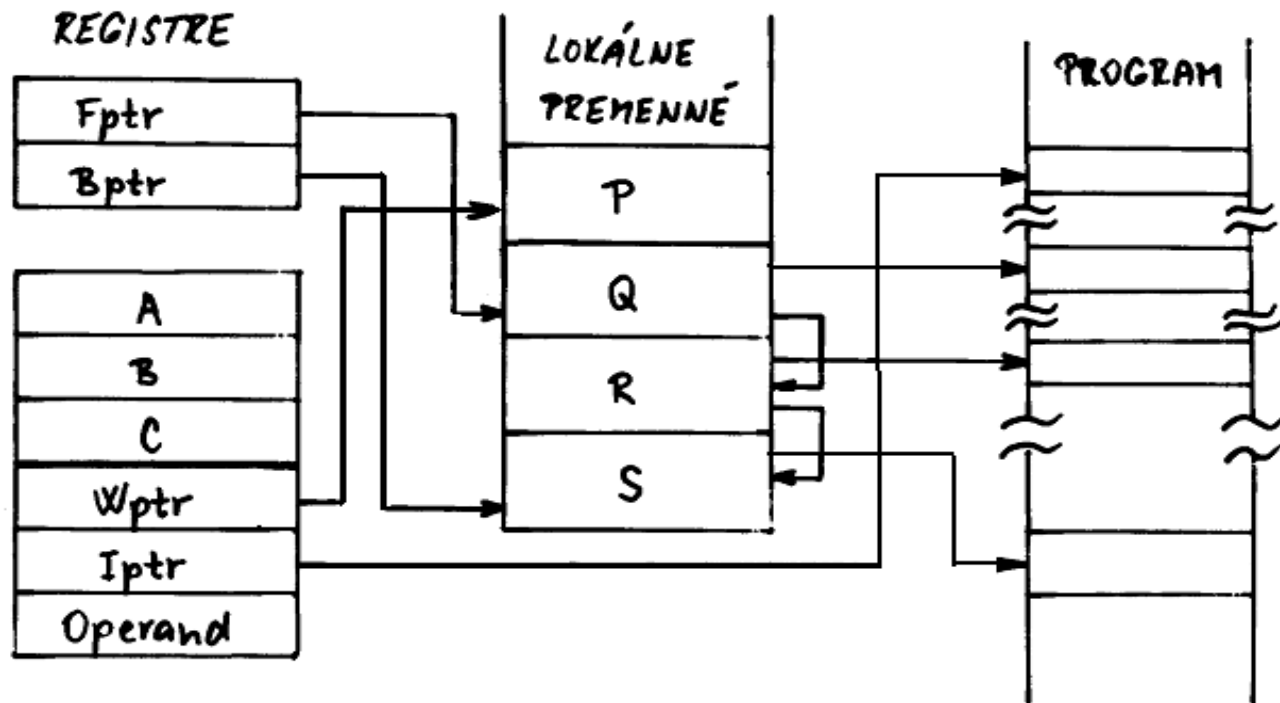
HW IMPLEMENTÁCIA: TRANSPUTER

- BEŽIACI PROCES S

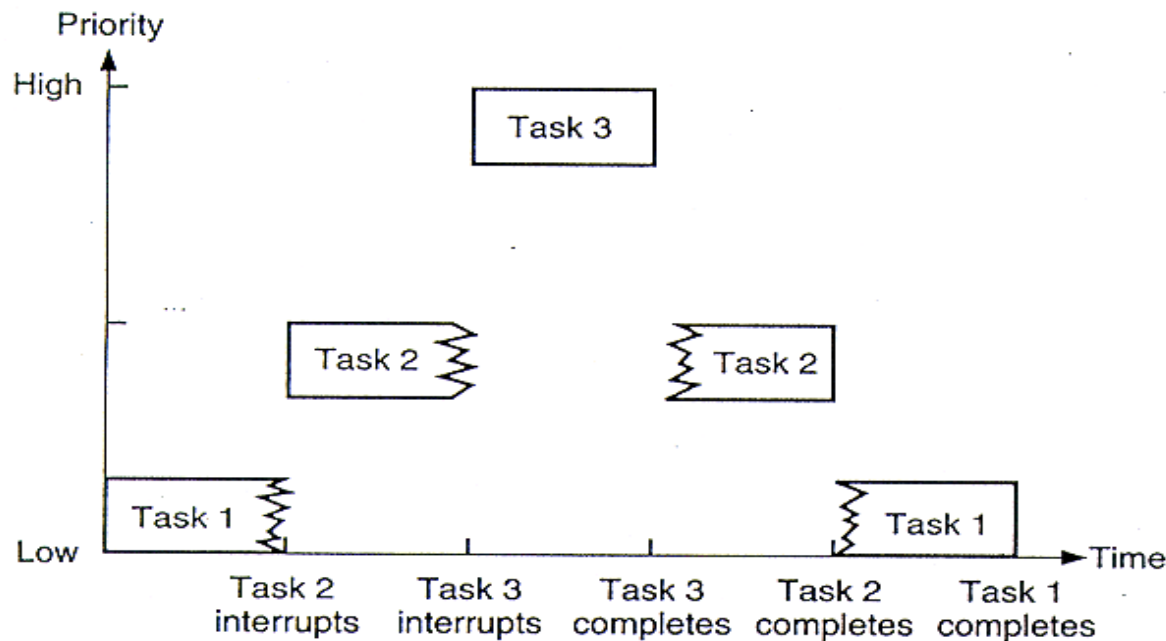


HW IMPLEMENTÁCIA: TRANSPUTER

- BEŽIACI PROCES P



METÓDY PLÁNOVANIA PROCESOV



- **PRIORITNÉ PLÁNOVANIE S PREEMPCIOU (x86 = TASK SWITCH)**

PLÁNOVATELNOST PROCESOV

NEZÁVISLÉ PERIODICKÉ PROCESY

- **T = perióda**
- **C = CPU time**
- **VYUŽITIE PROCESORA:**
(CPU Utilization)
 $U = C / T$

PLÁNOVATELNOST PROCESOV

- **TEORÉM 1.**
(Utilization Bound Theorem)
- **PRIORITY PROCESOV - RATE
MONOTONIC FUNCTION**

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_n}{T_n} \leq n \left(2^{\frac{1}{n}} - 1 \right) = U_{(n)}$$

PLÁNOVATELNOST PROCESOV

n	U(n)
1	1,000
2	0,830
3	0,779
4	0,756
5	0,743
6	0,734
7	0,728
8	0,724
9	0,720
∞	0,690

PLÁNOVATELNOST PROCESOV

- **Príklad 1.**

$$t_1: C_1 = 20; T_1 = 100; U_1 = 0,2$$

$$t_2: C_2 = 30; T_2 = 150; U_2 = 0,2$$

$$t_3: C_3 = 60; T_3 = 200; U_3 = 0,3$$

$$\underline{U(3) = 0,7 < 0,779}$$

$$\text{ak } C_3 = 90; T_3 = 200; U_3 = 0,45$$

$$\underline{U(3) = 0,85 > 0,779}$$

PLÁNOVATELNOST PROCESOV

- **TEORÉM 2**
(Completion Time Theorem)

**KONTROLA SPLNENIA MDO PRI
SÚČASNOM ŠTARTE VŠETKÝCH
PROCESOV**

PLÁNOVATELNOST PROCESOV

- **NEZÁVISLÉ PERIODICKÉ A
APERIODICKÉ PROCESY**
- **T_a , C_a , $U_a = C_a / T_a$**
- **ZNÍŽENIE VYUŽITIA PROCESORA –
REZERVOVANÝ ČAS NIE JE VŽDY
VYUŽITÝ**
- **PRIORITY APERIODICKÝCH PROCESOV**

PLÁNOVATELNOST PROCESOV

SPOLUPRACUJÚCE PROCESY

- **INVERZIA PRIORITY**
- **PRIRADENIE PRIORÍT**
- **GENERALIZED UTILIZATION BOUND
THEOREM**

$$U_i = \sum_{j \in H_0} \frac{C_j}{T_j} + \frac{1}{T_i} \left(C_i + B_i + \sum_{k \in H_1} C_k \right)$$

PLÁNOVATELNOST PROCESOV

- **Príklad 2.**

Per. t_1 : $C_1 = 20$; $T_1 = 100$

Per. t_2 : $C_2 = 15$; $T_2 = 150$

Per. t_3 : $C_3 = 30$; $T_3 = 300$

Aper. t_a : $C_a = 4$; $T_a = 200$

Priorita: t_a t_1 t_2 t_3

WCU pre každú úlohu

VNORENÉ SYSTÉMY SO ZVÝŠENOU SPOĽAHLIVOSŤOU

- **SPOĽAHLIVOSŤ (IEC 50-191):**

**SCHOPNOSŤ SYSTÉMU PLNÍŤ
POŽADOVANÉ FUNKCIE PRI ZACHOVANÍ
HODNÔT PREVÁDZKOVÝCH
UKAZOVATEĽOV V DANÝCH MEDZIACH A
V ČASE PODĽA STANOVENÝCH
TECHNICKÝCH PODMIENOK**

ZÁKLADNÉ POJMY

- ZLYHANIE
- PORUCHA:
TRVALÉ PORUCHY
OBČASNÉ PORUCHY
PRECHODNÉ PORUCHY
- CHYBA

ZÁKLADNÉ UKAZOVATELE SPOĽAHLIVOSTI

- PRAVDEPODOBNOŠŤ
BEZPORUCHOVEJ PREVÁDZKY $R(t)$
- PRAVDEPODOBNOŠŤ PORUCHY $Q(t) =$
 $1 - R(t)$
- HUSTOTA PRAVDEPODOBNOŠTI
PORÚCH $f(t) = dQ(t)/dt$

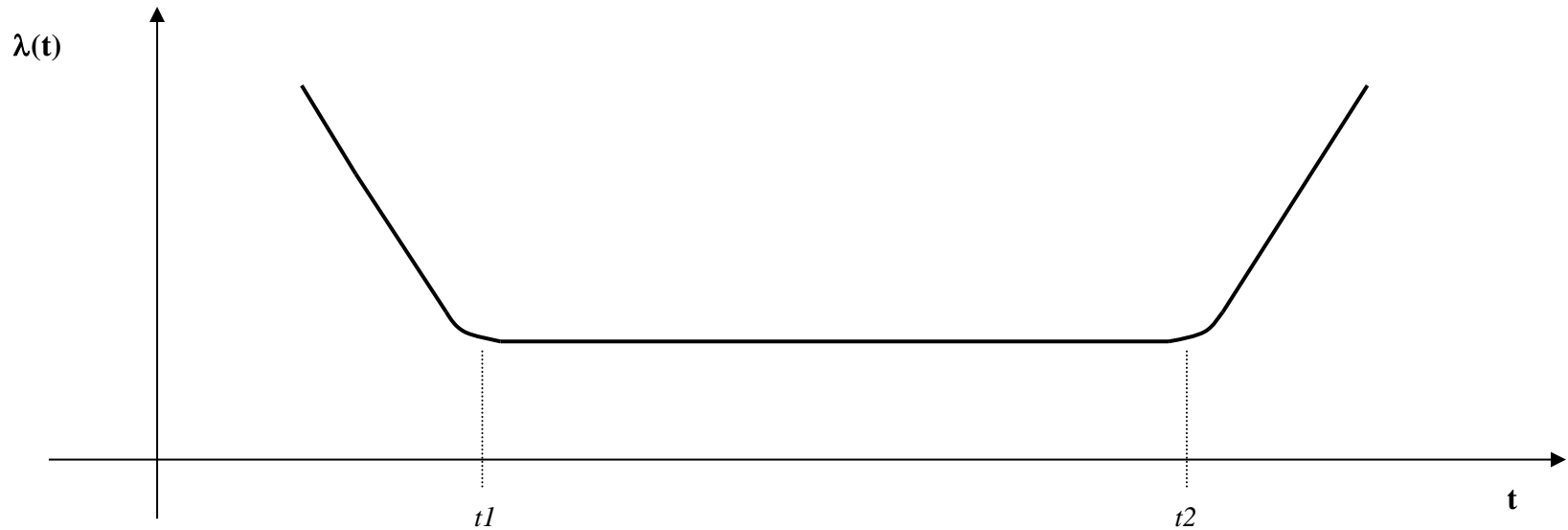
ZÁKLADNÉ UKAZOVATELE SPOĽAHLIVOSTI

- **INTENZITA PORÚCH**

$$\lambda(t) = f(t)/R(t)$$

- **STREDNÁ DOBA BEZPORUCHOVEJ
PREVÁDZKY T_S (*MTTF, MTBF*)**
- **STREDNÁ DOBA DETEKcie PORUCHY
*MTTD***
- **STREDNÁ DOBA OPRAVY *MTTR***

SÚVISLOSŤ UKAZOVATEĽOV SPOĽAHLIVOSTI



$$R(t) = e^{-\lambda \cdot t}$$

ZVYŠOVANIE SPOĽAHLIVOSTI SYSTÉMOV

- **PREDCHÁDZANIE PORUCHÁM**
- **TOLEROVANIE PORÚCH (PORUCHA SÚČIASTKY A PORUCHA SYSTÉMU)**
- **VPLYV TOLERANCIE PORÚCH NA UKAZOVATELE SPOĽAHLIVOSTI**

SYSTÉMY ODOLNÉ PROTI PORUCHÁM

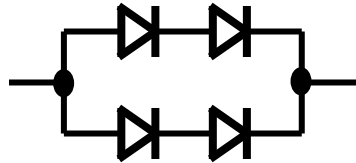
- **PROGRAM NIE JE ZASTAVENÝ ALEBO ZMENENÝ V DÔSLEDKU PORUCHY**
- **VÝSLEDOK JE SPRÁVNY**
- **VÝPOČET SKONČIL V PREDPÍSANOM ČASE**

SYSTÉMY ODOLNÉ PROTI PORUCHÁM

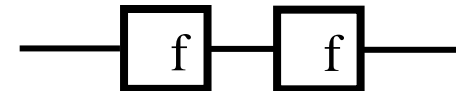
- **ODOLNOSŤ PROTI PORUCHÁM –
REDUNDANCIA**
- **FORMY:**
- **PRIESTOROVÁ (ZÁLOHA)**
- **ČASOVÁ**

ZÁLOHOVANIE

- STATICKÁ ZÁLOHA:
- ÚROVEŇ SÚČIASTOK

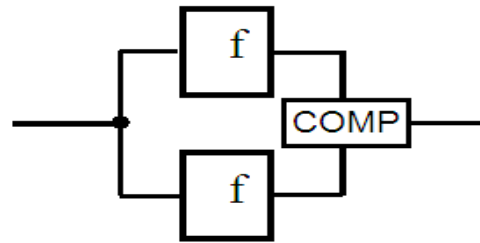


- ÚROVEŇ OBVODOV:
SÉRIOVÉ SPOJENIE PRVKOV

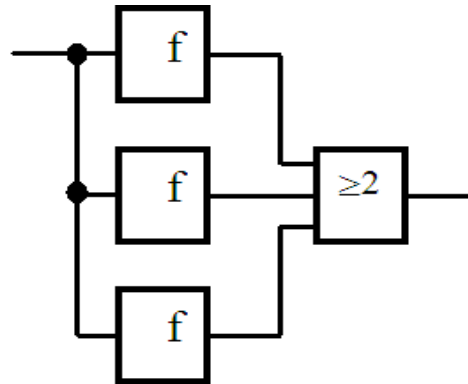


ZÁLOHOVANIE

PARALELNÉ SPOJENIE PRVKOV



MAJORITA

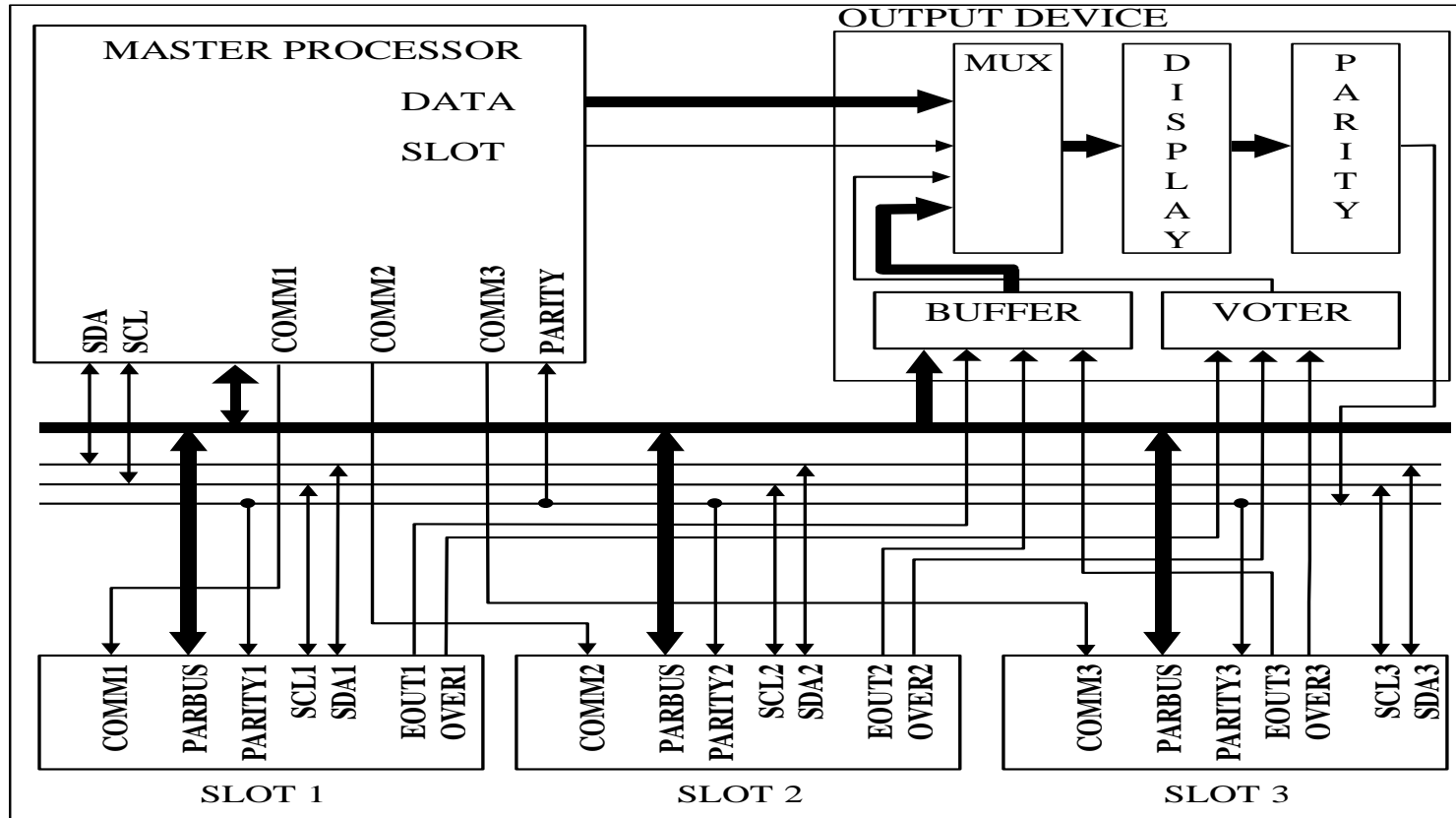


MAJORITY ČLEN

- **ZÁLOHOVANIE MAJORITY ČLENA**
- **ZÁLOHOVANIE SENZOROV**
- **ZÁLOHOVANIE AKČNÝCH ČLENOV**
- **REALIZÁCIA MAJORITY ČLENA**

ZÁLOHOVANIE

– ÚROVEŇ SYSTEMU



ZÁLOHOVANIE

- **DYNAMICKÁ ZÁLOHA**
 - NEZAŤAŽENÝ REŽIM
 - ODĽAHČENÝ REŽIM
 - ZAŤAŽENÝ REŽIM

- **HYBRIDNÁ ZÁLOHA**
 - $GMR = TMR + DZ$

SÚBEŽNÁ KONTROLA

DETEKCIA CHÝB NA ÚROVNI OBVODOV

- **KONTROLNÉ A SAMOOPRAVNÉ KÓDY
PRE PAMÄŤ**
- **SAMOČINNE KONTROLOVANÉ
OBVODY**

SÚBEŽNÁ KONTROLA

DETEKCIA CHÝB NA SYSTÉMOVEJ ÚROVNI

- SÚBEŽNÁ KONTROLA TOKU PROGRAMU
- KONTROLA PRIJATEĽ. VÝSLEDKOV
- KONTROLA PRÍSTUPOVÝCH PRÁV
- KONTROLA TYPU PAMÄTE
- KONTROLA PLATNOSTI OP. KÓDU

SÚBEŽNÁ KONTROLA

HW PODPORA DETEKČIE CHÝB

- PRIVILEGOVANÝ REŽIM x86