

Laborator - 25% nota finală  
- care nu are lab. este prezent în turul 1 (sesiunea ian.-feb.)  
și sunt calificat în turul 2. (sesiunea urm.)

Examen - proba scrisă - 6 subiecte (2 p. of.) (num. 5)  
- ~~proba orală~~  
    (1-4) teorie și probleme în limbajul C (cât un pct.)  
    (5-6) sub de programare (cât 2 pct.)  
Dacă lucră și faci a rezolva 5-6 probleme la proba orală, un sub suplimentar din C (simplu)  
- proba orală - teorie (subiecte simple)  
    2 subiecte - să știi ce ce se introduce

→ media -  $(\text{lab} + \text{scris} + \text{oral 1} + \text{oral 2}) / 4$  - notele se rotunjesc în fav. noastră  
turul 1 - dacă avem mix și în examen putem renunța la oral:  $(\text{lab} + 3 \cdot \text{scris}) / 4$   
Lucrările se ~~afixează~~ corectate în foia noastră,  
→ Restante - lab nu mai este determinant  
    - o în lab nu mai garantează promovarea.

### Bibliografie:

Programare:

1. A. Baranga - Dezvoltarea aplicațiilor C/C++ în sistemul de operare UNIX, Ed. Albatra Cluj-Napoca. - poate fi folosită la examen  
    o [www.albatra.ro](http://www.albatra.ro) - 25% discount  
    o Librăria Lumea (Biblu)

! La examen avem voie cu o foaie A4 - cu informații - apeluri sisteme

2. A. Baranga - Introducere în sistemul de operare UNIX, Ed. Albatra Cluj-Napoca - conține comenzile UNIX-ului, nu programare

Teorie:

3. M. Vlădu - Sisteme de operare  
    [www.unibuc.ro](http://www.unibuc.ro) - biblioteca virtuală a universității
4. Andrew Tanenbaum - Sisteme de operare moderne
5. A. Baranga, G. Enache - Sisteme de operare  
    [www.fmi.unibuc.ro](http://www.fmi.unibuc.ro)



• Ierarhia după Rumbaugh:

Nivele de organizare a arhitecturii calculatoarelor

- ① Fizic → hardware
- ② Microprogramat → ce instrucțiuni stie să facă hardware-ul cablat, - ce stă să facă hardware-ul, procesorul.

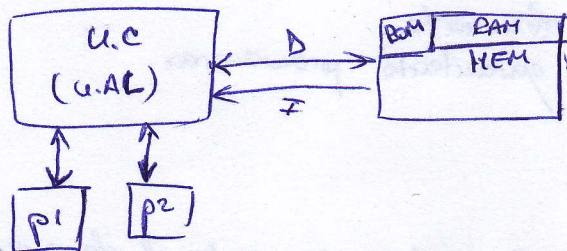
③ Limbaj de asamblare

④ Sistem de operare

⑤ Aplicație - utilizatori < <sup>casnic</sup> programatori

• J. Forthoferman

3 componente principale

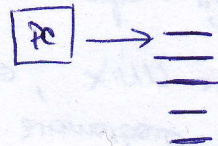


UC = unitatea centrală (mat. ceaspe) ← "cei care gâduresc"  
 UAL = unitatea aritmetică și logică  
 MEM = memorie (mem. albă) ← "folosita de structuri"  
 P1, P2 = periferice (de intrare / ieșire)  
 D = canalul de date  
 F = canalul de felci  
 ROM = Read Only Memory

• Fiecare procesor trebuie să aibă 2 registre cu anumite funcționalități

- Stack Pointer (nu există stivă violă)
- Program Counter (pointer) - nu are voie să aibă val aleatoare

PC = program counter



PC este singurul registru care nu are voie să aibă val aleatoare.

• Întreperii (hard)

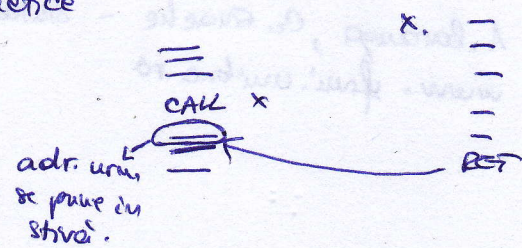
IR0, IR1, ..., IRn - nivele de întreperii

- intr. de date, calcule aritmetice
- intr. de tipul CALL și RET

CALL - jump și push

RET. - se ia din stivă și se efectuează în cont.

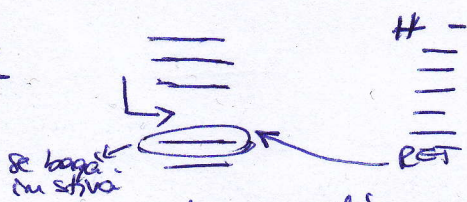
(se întoarce)





Complemente  
de arch. calc.

Interrupere



Fiecare nivel de intr. posedă un handler de tratare a interrupției

- Toți regiștrii pe care îi folotesc trebuie refăcuți înainte de PST (mecanismul push și pop în aceeași ordine (același nr de push-pop))

- O intr. de nivel superior nu poate fi întreruptă de una mai puțin importantă.

Prioritatea  $IR0 > IR1 > \dots$

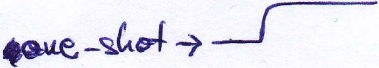
- Când avem o întrerupere, întreruperea induce pending întreruperea este ascunsă

Când se deactivază atunci trecem la rezolvarea ei.

- Circuit de tip ceas



Avem posibilitatea să primim o întrerupere la ceas.



- Funcționalitățile S.O. 8

Orice S.O. trebuie să îndeplinească obligatoriu:

1. Interfață utilizator
2. Gestionare periferice
3. Gestionare fișiere (pe disk) - apelează 2 (sunt periferice speciale)
4. Apeluri sistem.

Există 2 mari familii de S.O.: - windows  
- Unix

Funcționalități evaluate - se pot uita.