

Curs SO: <http://staff.cs.upt.ro/~danc/so/curs/so-curs.v.1.0.pdf>

Laburi SO:

<https://drive.google.com/file/d/18rvvghaJi03I5liU4O6SOK1TqTOupsjF/view?usp=sharing>

Pentru regex: <https://regexr.com/>

Subiecte vechi SO:

<https://drive.google.com/drive/folders/1Kp1VaK0NYZ1MApTtz5yczAFFzMkv6zlx?usp=sharing>

(sunt si unele vechi care nu sunt grila, nu sunt importante banuiesc din moment ce acum e grila)

Negru -> raspunse + verificate

Albastru -> neraspunse

Portocaliu -> deja raspunse, NU SUNT VERIFICATE

Rosu -> raspunse, nu e sigur (nu stergeti si restul raspunsurilor ca sa putem verifica ulterior)

## Intrebari Romana (unul sau mai multe rasp corecte)

1. Comanda chmod 512 x va avea urmatorul efect pentru fisierul x:

- Activeaza dreptul de read pentru proprietar
- Activeaza dreptul de executie pentru grup

- **r** (read) = 4
- **w** (write) = 2
- **x** (execute) = 1
- no permissions = 0

Restul variantelor:

- Activeaza dreptul de write pt proprietar
- Activeaza dreptul de read pentru grup
- Activeaza dreptul de executie pentru ceilalti utilizatori

2. Functia kill()

- Trimite un semnal

Restul variantelor:

- Termina intotdeauna un proces, in mod brusc
- Distruge fara drept de apel un proces
- Capteaza un semnal
- Trimite semnalul cu numarul 9 catre comanda curenta

3. Informatiile despre numarul de legaturi spre un fisier se gasesc:

- In directorul curent
- In directorul parinte
- In nodul index
- In intrarea in director
- In campul `st_mode`

4. Legaturile fixe sunt:

- Intrari in director diferite care refera acelasi nod index

Restul variantelor:

- Nume alternative pentru acelasi fisier, aflate intotdeauna intr-un acelasi director
- Fisierile individuale care refera alte fisiere
- Fisierile individuale care refera directoare fixe
- Fisierile speciale

5. Comanda `grep -E '^.[st].+\.[t[emp]xt]*$'` primind pe cate o linie sirurile de mai jos le va afisa la iesirea standard pe urmatoarele:

- `Conf.t` - incorrect
- `S1.text` - incorrect
- `Config.tmp` - incorrect
- `S1.ttm` - incorrect
- `os1.text` (Raspuns corect)

6. Comanda `grep 'a.[o-s]$. $'` primind pe cate o linie sirurile de mai jos le va afisa la iesirea standard pe urmatoarele:

- `axpz`
- `megafonul`(nu satisface regexul - verificat)
- `rtatry`
- `afon`
- `megafon`

7. Codul de mai jos va afisa la iesirea standard (alegeti toate variantele corecte)

```
int main (int argc, char ** argv)
{
    int pid1, pid2;
    If ((pid1=fork())<0)
    {
        printf("Error\n");
        exit(1);
    }
    If (pid1!=0)
        printf("A\n");
    printf("B");
    If ((pid2=fork())<0)
    {
        printf("Error\n");
        exit(2);
    }
    printf("A\n");
    Return 0;
}
Rezultat : ABABABABA
```

Daca e sa o luam logic ar trebui sa se afiseze B doar de doua ori. Se afiseaza de 4 pt ca nu are '\n' la final si e o anomalie la fork() (fara '\n' la final ce ai in printf ti se pune intr-un buffer care se sterge la urm print deci ti se copiaza si buffer-ul ala in fork => de aia e de 4 ori)

E POSIBIL CA RASP CORECTE CONSIDERATE SA FIE 5 A SI 2 B DIN MOM CE NU CRED  
CA NI S-A SPUS DE ASTA/A COMPILAT PROFUL CODUL

- 3 caractere B
- 2 caractere B
- 2 caractere A
- 5 caractere A
- 1 caracter B

8. Marcati pe foaia de raspuns numai afirmatiile adevarate de mai jos:

- Sistemul de operare ruleaza exclusiv in spatiul utilizator(user space) - Incorect
- Memoria de date a unui proces este partajata intre toate firele de executie din sistem
- Un proces UNIX poate contine mai multe fire de executie
- Sistemul de operare ofera o interfata de programare pentru aplicatii
- Stiva e partajata intre firele de executie - Incorect

9. Java Virtual Machine este o masina virtuala:

- Tip proces - Verificat

Restul variantelor:

- Tip sistem
- Container
- Implementata in interiorul nucleului sistemului de operare UNIX
- Specifica UNIX

10. Codul unei functii de biblioteca se afla:

- In nucleul sistemului de operare
- Intr-un header C
- Intr-un fisier binar
- In partitia de swap
- In varful stivei

11. Un televizor non-smart, care poate rula filme de pe un disc USB are sistem de operare?

- Da, pentru ca e un dispozitiv electronic
- Da, pentru ca e un sistem de calcul cu procesor si dispozitive periferice
- Nu, pentru ca e deservit de un firmware
- Nu, pentru ca este un dispozitiv electronic de uz general

- Nu, pentru ca e format din componente proprietare

12. fork() este:

- Un apel care creeaza un proces - Verificat
- O comanda care creeaza un proces nou si returneaza identificatorul său in parinte - Verificat (nush daca e comanda)

Restul variantelor:

- O comanda care creeaza doua procese
- O functie care creeaza doua proceste distincte
- Un apel care creeaza un proces fiul si unul parinte

13. Daca un proces trimite cu succes date unui alt proces printr-un pipe, inseamna in mod necesar ca:

- Procesele sunt fiii aceluiasi proces distinct
- Procesele au mostenit capetele necesare ale pipe-ului
- Exista un proces tert care le este stramos comun celor doua procese
- Procesele au inchis capetele nefolosite ale pipe-ului
- Procesele nu au inchis capetele utile ale pipe-ului la care au acces

14. Daca un semnal e trimis unui proces atunci:

- Daca procesul trateaza semnalul si acest lucru e posibil, sa apleleaza o functie din acel proces
- Daca procesul nu trateaza semnalul este intotdeauna terminat
- Procesul poate desemna un proces scris special pentru a trata semnalul
- Procesul poate sa ignore semnalul indiferent de caz
- Procesul trimite automat un semnal tuturor proceselor fiu

15. Informatiile despre drepturile de acces la un fisier se gasesc:

- In nodul index asociat acestuia

Restul variantelor:

- In sistemul de operare
- In fisier
- In directorul care contine fisierul
- In legatura simbolica

16. Numarul de blocuri de date ocupate de un fisier pe disc se poate afla:

- Insumand informatiile din tabela de blocuri ocupate, aflata in nodul index

Restul variantelor:

- Parcurgand primele 10 intrari din tabela blocurilor ocupate
- Citind cele trei intrari de indirectare din tabela de blocuri
- Afland adresa pe disc a ultimului bloc ocupat de fisier
- Citind o informatie din nodul index si aplicand o formula care depinde si de dimensiunea blocului

17. Un router care poate fi administrat folosind un browser Web are sistem de operare?

- Da, pentru ca e un dispozitiv electronic
- Da, pentru ca e un sistem de calcul cu procesor si dispozitive periferice
- Nu, pentru ca e deservite de un firmware
- Nu, pentru ca este un dispozitiv electronic de uz general
- Nu, pentru ca e format din componente proprietare

18. Linia urmatoare:

```
Echo "cat f \ | wc -l" > g
```

- Scrie intr-un fisier dimensiunea (in octeti) unui alt fisier
- Afiseaza pe ecran dimensiunea unui fisier
- Compara numarul de linii dintr-un fisier cu o variabila g
- Scrie intr-un fisier numarul de linii din alt fisier
- Scrie intr-un fisier sirul de caractere cat f | wc -l

19. Planificarea la executie a proceselor este:

- O functionalitate implementata de sistemul de operare

Restul variantelor:

- Un apel sistem
- O comanda care planifica executia programelor la diferite momente de timp
- O facilitate existenta in toate interpretoarele de comenzi moderne
- Un algoritm de gestiune a drepturilor de executie pentru programe
- 

20. Un program C primeste parametrii din linia de comanda in variabila argv. Daca e compilat si apelat ca mai jos, care e argv[3]?

```
./program "1 2" "2 1" 2 3 '1' 2 3
```

- 1
- 2
- 1 2
- 3 1
- 3

21. Codul de mai jos va afisa la iesirea standard (alegeti toate variantele corecte)

```
int main (int argc, char ** argv)
{
    int pid1, pid2;
    If ((pid1=fork())<0)
    {
        printf("Error\n");
        exit(1);
    }
    printf("C\n");
    If ((pid2=fork())<0)
    {
        printf("Error\n");
        exit(2);
    }
    printf("B\n");
    Return 0;
}
```

- 4 caractere B
- 1 caracter B
- 2 caractere C
- 1 caracter C
- 3 caractere B

22. Un proces A creeaza un nou proces B, dupa care procesul A apeleaza cu succes functia `execvp()` pentru a porni un program P. Care vor fi consecintele?

- Procesul A nu va mai putea prelua explicit valoarea returnata la terminare de B
- Procesul care ruleaza P va avea acelasi PID ca A

Restul variantelor:

- Procesul A va fi terminat complet si un nou proces va rula un program de pe disc
- Procesul care ruleaza P va avea alt PID decat A

- B se termina automat pentru ca nu mai are parinte

23. Diferenta intre `execvp()` si `execl()` este

- `execvp()` specifica o cale, dar `execl` nu
- `execvp()` tine cont de date memorate intr o anumita variabila de mediu, spre deosebire de `execl` care nu foloseste acea variabila
- `execvp()` lanseaza un proces, pe cand `execl` nu
- `execvp()` lanseaza un proces, pe cand `execl()` executa un fisier de comenzi dat
- `execvp()` este nerecomandat a fi folosit, spre deosebire de `execl()`

24. Functia `alarm()`

- Programeaza trimiterea unui semnal anume in viitor

Restul variantelor:

- Anunta procesele ca trebuie sa se termine imediat
- Semnaleaza o situatie de eroare aparuta la executie
- Reseteaza orice alarma setata anterior in acelasi proces
- Informeaza un proces despre aparitia unei erori in alt proces

25. Un proces este:

- Un program sau o parte de program aflat(a) sub supervizarea sistemului de operare
- O comanda interna
- Un program C care apeleaza `fork()`
- Un serviciu oferit de sistemul de operare programelor
- O functie de sistem care implementeaza serviciul de multitasking

26. `open()` este:

- Un apel sistem
- O comanda care deschide un fisier
- O functie de biblioteca
- O functie aflata in interfata de programare oferita de UNIX
- O comanda care este implementata in nucleul sistemului de operare

27. Marcati in foaia de raspuns numai afirmatii adevarate de mai jos:



- Sistemul de operare are acces la spatiul nucleu(kernel space)
- Sistemul de operare ofera o interfata de programare pentru aplicatii
- Stiva e private fiecarui proces
- Memoria de date a unui proces este partajata intre toate firele de executie din sistem
- Un proces UNIX poate contine mai multe fire de executie

28. Cand poate functia read() returna un numar mai mic decat al treilea parametru primit?

- cand se citește ultimul set de date dintr-un fișier
- in caz de eroare
- cand fisierul citit e mai mic, in bytes, decat numărul de octeti ceruti

Restul variantelor:

- Cand se citesc date dintr-un pipe si procesele conectate la capatul de scriere nu mai furnizeaza date
- Cand in fisier mai exista necititi un numar mai mare de bytes decat cel cerut

29. La terminare, un program returneaza in mod obligatoriu catre apelant:

- Un numar intreg

Restul variantelor:

- O valoare intreaga la iesirea standard
- Un numar natural la iesirea standard
- Un sir de caractere
- Un numar real la iesirea standard

30. Un proces aflat in executie care apeleaza read() pe capatul de citire dintr-un pipe in care nu sunt date va fi trecut in starea:

- blocat

Restul variantelor:

- Pregatit de executie

- In executie
- Trimis in swapping
- Terminat

31. Iesirea standard se poate redirecta catre un fisier pentru care exista un descriptor valid fd, astfel:

- `dup2(fd,1)`

Restul variantelor:

- `open(1, fd);`
- `redirect(1, fd);`
- `redirect(fd, 1);`
- `dup2(1, fd);`

32. O variabila de mediu bash

- e de tipul sir de caractere

Restul variantelor:

- Este un numar intreg
- Este un numar natural
- Este un numar real
- Poate avea orice tip

33. Comanda `wc -l` numara:

- liniile primite la intrarea standard

Restul variantelor:

- Coloanele primite la intrarea standard
- Caracterele primite la intrarea standard
- Liniile scrise la iesirea standard
- Coloanele scrise la iesirea standard

34. `/dev/null` este:

- Un fisier special spre care orice date sunt scrise se pierd

Restul variantelor:

- Un pipe cu nume

- Un fisier special din care se citesc tot timpul valori nule
- Un fisier special din care se citesc valori aleatoare
- Un fisier special care indica spre un dispozitiv periferic deconectat temporar

35. Nucleul unui sistem de operare bazat pe arhitectura microkernel:

- Oferă servicii de bază, cum ar fi gestiunea proceselor
- E parte a unei arhitecturi flexibile, cu mentenanță ușoară
- Este format din mai multe servere independente care oferă servicii

Restul variantelor:

- Are dezavantajul că ocupă singură toată memoria
- Este un concept pur teoretic, care nu a fost niciodată implementat tehnic

36. Bash este:

- Un shell
- Un interpretor de comenzi

Restul variantelor:

- Un shell script
- Un apel de sistem
- O funcție de bibliotecă

37. Un blu-ray player care poate accesa servicii de streaming (Youtube, Netflix, etc) are sistem de operare?

- da, pentru că e un sistem de calcul cu procesor și dispozitive periferice

Restul variantelor:

- Da, pentru că e un dispozitiv electronic
- Nu, pentru că e deservit de un firmware
- Nu, pentru că este un dispozitiv electronic de uz general
- Nu, pentru că este format din componente proprietare

38. Un fisier poate fi făcut disponibil pentru două procese astfel:

- un proces creează fisierul și apoi îl creează pe celălalt proces
- ambele procese cunosc descriptorul de tip `int` al fisierului

Restul variantelor:

- Procesul care deschide fisierul trimite descriptorul de fisier obținut la `open()` celuilalt proces

- Un proces trimite catre celalalt numele fisierului
- Se foloseste functia fdopen care trimite un identificator tip FILE\* al fisierului

39. Daca un proces trimite intr-un pipe o structura de date:

- Un alt proces poate citi doar jumătate din structura, in anume conditii
- Poate exista un proces care poate citi toata structura

Restul variantelor:

- Un proces e obligat sa citeasca toata structura, altfel primeste cod de eroare
- orice proces din sistem poate citi structura
- Apare intotdeauna eroare

40. In UNIX, un fir de executie poate, daca este programat in acest scop:

- Termina procesul curent
- Apela functii
- Returna o valoare

Restul variantelor:

- Accesa memoria oricarui alt fir de executie care ruleaza pe calculator
- Defini un set de variabile globale accesibile doar lui insusi

41. Un programator doreste sa transmita un sir de dimensiune variabila mai mica decat 100, prin intermediul unui pipe. Sirul este continut intr-o variabila char s[100] si a fost initializat apeland strcpy(), iar descriptorii asociati pipe-ului se afla in variabila pfd, pentru a scrie corect, in acest scop, sirul in pipe, trebuie apelat

- write (pfd[1],s,strlen(s));

Restul variantelor:

- write(pfd[0], s, strlen(s));
- write(pfd[0], &s, strlen(s)); (nu vad sigur daca e &s)
- write(pfd[1], s, sizeof(s));
- write(pfd[1], &s, strlen(s));

42. Un program C primeste parametrii din linia de comanda in variabila argv. Daca e compilat si apelat ca mai jos, care e argv[2]?

./program "1 2" "2 1" 2 3 '1' 2 3

- 2 1

## Intrebari Engleza

1. The UNIX operating system provides as specific functionalities:

- Memory management
- Application programming interface
- Database management
- Website management
- Process management

2. When having one process A that creates another process B by calling fork(), after process B ends, while process A does not call wait to retrieve the status of B, process B is:

- Parent process
- Orphan process
- Daemon process
- Child process
- Zombie process

3. SETUID and SETGID flags refer to:

- Files on the file system
- The saved execution privileges of a running process SAU
- The rights for modifying a file
- The past execution privileges of a process
- The current execution privileges of a process

4. A child process inherits from its parent:

- All the values on the stack as they were before the process was created
- The process identifier
- All global variables
- All file descriptor redirections
- All open files

5. Select the correct code sequences among the following options when reading a text file in a buffer

```
#define BLOCK_SIZE 4096
char buffer[BLOCK_SIZE];
Int fd = open(file_name, O_RDONLY);
```

- While ((n=read(fd, buffer, BLOCK\_SIZE)) > 0)  
{  
    buffer[n]='\0';  
    for(int i=0; i<strlen(buffer); i++) {...}  
}
- While (read(fd, buffer, BLOCK\_SIZE) > 0)  
{  
    for(int i=0; i<strlen(buffer); i++) {...}  
}
- While ((n=read(fd, buffer, BLOCK\_SIZE)) > 0)  
{  
    for(int i=0; i<strlen(buffer); i++) {...}  
}
- While (read(fd, buffer, BLOCK\_SIZE) > 0)  
{  
    for(int i=0; buffer[i]; i++) {...}  
}
- While ((n=read(fd, buffer, BLOCK\_SIZE)) > 0)  
{  
    for(int i=0; i<n; i++) {...}  
}

6. What will the output be for the following sequence?

```
var=23
Echo ``expr $var + 3``
Echo ``expr $var + 3``
```

- 26  
  `expr \$var + 3`
- `expr \$var + 3`  
  `expr \$var + 3`
- `expr \$var + 3`  
  26
- 26  
  26

7. To find out what path a symbolic link refers to, and the link's size on disc, a programmer needs to call

- readlink
- ln
- unlink
- lstat
- stat

8. When having one process A that creates another process B by calling fork(), process A is:

- Parent process
- Orphan process
- Daemon process
- Child process
- Zombie process

9. If a shell script is called as follows:

```
sh script.sh 1 2 3
```

What will the output be for the following command?

```
Echo $0 $1 $2
```

- 1 2 3
- Script.sh 1 2
- Sh script.sh 1
- Error message

10. The correct call(s) – with valid argument only – to create a *file* to later write in is :

Apelurile corecte pentru crearea unui fisier in care se va scrie ulterior sunt:

- creat(file, S\_IRUSR | S\_IWUSR);
- open(file, O\_WRONLY | O\_CREAT | O\_TRUNC | S\_IRUSR | S\_IWUSR);
- open(file, O\_WRONLY | O\_CREAT | O\_TRUNC );
- open(file, O\_WRONLY | O\_CREAT | O\_TRUNC, S\_IRUSR | S\_IWUSR );
- creat (file, O\_WRONLY | O\_CREAT | O\_TRUNC);

