

Subiecte OS

1. Marcăți afirmațiile adevărate de mai jos:

- a) Sistemul de operare are acces la spațiul nucleu(kernel space)
- b) Memoria unui fir de execuție este partajată între toate procesele din acel fir
- c) Un proces UNIX poate fi format din mai multe procese
- d) Sistemul de operare oferă o interfață de programare pentru aplicații
- e) Stiva este privată fiecărui proces

2. Când poate funcția `read()` returna un număr mai mic decât al treilea parametru primit?

- a) Când se citește ultimul set de date dintr-un fișier
- b) Când se citesc date dintr-un pipe și procesele conectate la capatul de scriere nu mai furnizează date
- c) În caz de eroare
- d) Când în fișier mai există necititi un număr mai mare de bytes decât cel cerut
- e) Când fișierul citit este mai mic, în bytes, decât numărul de octeți ceruți

3. La terminare, un program returnează în mod obligatoriu către apelant:

- a) Un număr întreg
- b) O valoare întreagă la ieșirea standard
- c) Un număr natural la ieșirea standard
- d) Un șir de caractere
- e) Un număr real la ieșirea standard

4. Planificarea la execuție a programelor este:

- a) Un apel sistem
- b) O comandă care planifică execuția programelor la diferite momente de timp
- c) O funcționalitate oferită de sistemul de operare

- d) O facilitate existenta in toate interpretoarele de comenzi moderne
- e) O facilitate oferita de toate shell-urile din UNIX
- f) Un algoritm de gestiune a drepturilor de executie pentru programe

5. 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:

- a) blocat
- b) pregatit de executie
- c) in executie
- d) trimis in swapping
- e) terminat

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

- a) `dup2(fd,1)`
- b) `open(1,fd)`
- c) `redirect(fd,1)`
- d) `redirect(1,fd)`
- e) `dup2(1,fd)`

7. O variabila de mediu `bash`:

- a) e de tipul sir de caractere
- b) este un numar intreg
- c) este un numar natural
- d) este un numar real
- e) poate avea orice tip

8. Comanda `wc -l` numara:

- a) liniile primite la intrarea standard
- b) coloanele primite la intrarea standard

- c) caracterele primite la intrarea standard
- d) liniile scrise la iesirea standard
- e) coloanele scrise la iesirea standard

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

- a) parcurgand primele 10 intrari din tabela blocurilor ocupate ???
- b) citind o informatie din nodul index si aplicand o formula care depinde si de dimensiunea blocului
- c) citind cele 3 intrari de indirectare din tabela de blocuri
- d) insumand informatiile din tabela de blocuri ocupate, aflata in nodul index
- e) afland adresa pe disc ultimului bloc ocupat de fisier

10. /dev/null este:

- a) un pipe cu nume
- b) un fisier special din care se citesc tot timpul valori nule
- c) un fisier special din care se citesc valori aleatoare
- d) un fisier special spre care orice date sunt scrise se pierd
- e) un fisier special care indica spre un dispozitiv periferic deconectat temporar

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

- a) este format din mai multe servere independente ce ofera servicii
- b) ofera servicii de baza, cum ar fi gestiunea proceselor
- c) are dezavantajul ca ocupa singur toata memoria ?
- d) e parte a unei arhitecturi flexibile, cu mentenanta usoara
- e) este un concept pur teoretic

12. bash este :

- a) un shell
- b) un shell-script

c) un apel sistem

d) un interpretor de comenzi

e) o functie de biblioteca

13. Un blu-ray player care poate accesa servicii de streaming (ex youtube,netflix,etc) are sistem de operare?

a) Da, pentru ca e un dispozitiv electronic

b) Nu, pentru ca e deservit de un firmware

c) Nu, pentru ca e un dispozitiv electronic de uz general

d) Da, pentru ca e un sistem de calcul cu procesor si dispozitive periferice

e) Nu, pentru ca e format din componente proprietare

14. Un fisier poate fi facut disponibil pentru 2 procese astfel :

a) Procesul care deschide fisierul trimite descriptorul de fisier obtinut la open() celui alt proces

b) Un proces trimite catre celalalt numele fisierului

c) Un proces creeaza fisierul apoi il creaza pe celalalt proces

d) Se foloseste functia fdopen() care trimite un indicator de tip FILE* al fisierului

e) Ambele procese cunosc descriptorul de tip INT al fisierului

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

a) Un proces e obligat sa citeasca toata structura, altfel primeste cod de eroare

b) Un alt proces poate citi doar jumatate din structura in anumite conditii

c) Poate exista un proces care poate citi toata structura

d) Orice proces din sistem poate citi structura

e) Apare intotdeauna eroare

16. Marcati afirmatiile adevarate

a) Sistemul de operare ruleaza exclusiv in spatiul utilizator (User space)

b) Memoria de date a unui proces e partajata intre toate firele de executie in sistem

c) Un proces UNIX poate contine mai multe fire de executie

d) Sistemul de operare ofera o interfata de programare pentru aplicatii

e) stiva e partajata intre firele de executie

17. fork() este :

a) o comanda care creeaza 2 procese

b) o functie care creeaza 2 procese distincte

c) un apel care creeaza un proces

d) un apel care creeaza un proces fiu si unul parinte

e) o comanda care creeaza un proces nou si returneaza identificatorul sau in parinte

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

a) activeaza dreptul de read pentru proprietar

b) activeaza dreptul de write pentru proprietar

c) activeaza dreptul de read pentru grup

d) activeaza dreptul de executie pentru grup

e) activeaza dreptul de executie pentru ceilalti utilizatori

19. Java Virtual Machine e o masina virtuala:

a) tip sistem

b) tip proces

c) container

d) implementata in interiorul nucleului sistemului de operare UNIX

e) e specifica UNIX

20. Informatiile despre drepturile de acces se gasesc:

a) in nodul index asociat acestuia

b) in fisier

- c) in directorul care contine fisierul
- d) in sistemul de operare
- e) in legatura simbolica

21. 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?

- a) procesul A va fi terminat complet si un nou proces va rula un program de pe disc
- b) programul nu va mai putea prelua explicit valoarea returnata la terminare de B
- c) procesul care ruleaza P va avea acelasi pid ca A
- d) procesul care ruleaza P va avea alt pid decat A
- e) B se termina automat pentru ca nu mai are parinte

22. Functia `kill()` :

- a) termina intotdeauna un proces in mod binar
- b) trimite un semnal
- c) distruge fara drept de apel un proces
- d) capteaza un semnal
- e) trimite semnalul cu numarul 9 catre comanda curenta

23. Functia `alarm()` :

- a) programeaza trimiterea unui semnal anume in viitor
- b) semnaleaza o situatie de eroare aparuta la executie
- c) reseteaza o alarma declansata
- d) informeaza un proces despre aparitia unei erori in alt proces
- e) anunta procesele ca trebuie sa se termine imediat

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

- a) termina procesul curent
- b) accesa memoria oricarui alt fir de executie care ruleaza pe calculator

c) defini un set de variabile globale accesibile doar lui insusi

d) apela functii

e) returna o valoare

25. Legaturile fixe sunt:

a) intrari la director diferite care refera acelasi nod index

b) nume alternative pentru acelasi fisier, aflate intotdeauna intr-un acelasi director

c) fisiere individuale care refera alte fisiere

d) fisiere individuale care refera directoare fixe

e) fisiere speciale

26. Functia fork() creeaza:

a) un program diferit de programul parinte, dar initial identic cu acesta

b) un fiu al programului curent, incarcat de pe disc

c) un nou proces cu acelasi identificator ca procesul curent

d) un proces nou

e) un proces parinte

27. Un programator dorește sa transmită un șir de dimensiune variabila < 100 prin intermediul unui pipe. Șirul e continut într-o variabila char s[100] și a fost inițializat apelând strcpy() iar descriptorii asociati pipe-ului se afla in variabila pfd[]. Pentru a scrie corect, in acest scop, sirul in pipe, trebuie apelat:

a) write(pfd[0], s, strlen(s));

b) write(pfd[0], &s, sizeof(s));

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

d) write(pfd[1], s, sizeof(s));

e) write(pfd[1], &s, strlen(s));

28. Informațiile despre numărul de legături spre un fișier se găsesc:

a) in directorul curent

b) in directorul parinte

c) in nodul index

d) la intrarea in director

e) in campul st_mode

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

-program "1 2" "3 1" "2 3" "1" 2 3

a) 1

b) 2

c) 1 2

d) 3 1

e) 3

30.The code for a library function resides:

a. In the OS kernel

b. In a C header

c. In a binary file

d. In the swap partition

e. At the top of the stack

31.A pipe works:

a. Based on the FIFO principle

b. Based on a LIFO principle

c. Based on the FILO principle

d. As a bidirectional data channel

e. As a unidirectional data channel

32.The target of a double indirection entry from the specific data block list stored in the i-node is :

a. Another i-node

b. A block with pointers to the other blocks

c. A data block of the file

d. Double file...(blocks/i-nodes)

e. A symbolik link to the file

33.A pipe must be created in a program:

a. Mandatory before creating all the processes

b. Before creating all the processes that need it

c. After the last fork() in the program

d. Immediately after creating the processes that use it

e. Only in processes that have already created a process

34.The UNIX OS provides:

a. Database management

b. Memory management

c. Application programming interface

d. Process management

..... Management??

35.What command can we use in UNIX to create a folder:

a. Mkf

b. Mkdir

c. Newdir

d. Mdir

e. Mkfolder

36.In UNIX a thread is able to

a. terminate the current process

b. Access the memory of all the threads running on the same computer

c. Define a set of global accessible only to itself

d. Call functions

e. Return a value

37. Examples of OS:

a.Windows

b.MS-DOS

c.IBM

d.Linux

38.In UNIX, user written programs:

a. Must get their data from the standard input

b. May send data to other programs ... provided by the OS

c. Can be interconnected using ... regardless of the way they are implemented ????????

d. Can redirect the standard output themselves

e.Can be made of several processes

39.The kernels of an OS based on the microkernel architecture :

a. Is made several independent servers that offer services

b. Offers basic services such as process management

c. Has the downside that it fills the entire memory of its own

d. Is part of a flexible architecture with easy maintenance

e. Is a purely theoretical concept that has never been implemented

40. The link count of a file is found in:

a. The working directory

b. The parent directory

c. The inode

d. The directory entry

e. The st_mode field

41. Sending the strings in the answer below , each on a separate line, to the command:

```
grep -E '[sf].+\.t[empxt]*$'
```

Will have the effect of printing at the standard output:

a. Conf.t

b. S1.txt

c. Config.tmp

d. S1.ttm

e. os1.text

42. Sending the strings in the answer below , each on a separate line, to the command:

```
grep -E '^[st].+\.t[empxt]*$'
```

Will have the effect of printing at the standard output:

a. conf.t

b. os1.text

c. config.tmp

d. s1.ttm

e. s1.text

43. Un router care poate fi administrat folosind un browser web are system de operare?

a. da, pentru ca este un dispozitiv electronic

b. da, pentru ca e un system de calcul cu processor si dispozitive periferice

c. nu, pentru ca e deservit d eun firmware

d.nu, pentru ca e un dispozitiv electronic de uz general

e. nu, pentru ca e format din component proprietare

44. Comanda : `echo "cat f\ | wc-l ">g`

- a. scrie in fisier dimensiunea (in octeti) anui alt fisier
- b. afiseaza pe ecran dimensiunea unui fisier
- c. compara numarul de linii dintr-un fisier cu o variabila g
- d. scrie intr-un fisier numarul de linii al unui alt fisier
- e. scrie intr-un fisier sirul de caractere cat f | wc-l

45.Codul de mai jos va afisa la iesirea standard:

```
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;  
}
```

- a. 4 caractere B
- b. 1 caracter B
- c. 2 caractere C
- d. 1 caracter C
- e. 3 caractere B

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

- a. da, pentru ca este un dispozitiv electronic
- b. da, pentru ca este un sistem de calcul cu dispozitive periferice
- c. nu, pentru ca e deservit de un firmware
- d. nu, pentru ca este un dispozitiv electronic de uz gneral

e. nu, pentru ca e format din componente proprietare

47. Daca un proces a trimis cu succes date unui alt proces printr-un pipe, inseamna in mod necesar ca:

a. procesele sunt fiii aceluiasi proces distinct

b. procesele au mostenit capetele necesare pipe-ului

c. exista un proces tert care le este stramos comun celor 2 procese

d. procesele au inchis capetele nefolosite ale pipe-ului

e. procesele nu au inchis capetele utile ale pipe-ului la care au acces

48. Daca un semnal e trimis unui proces, atunci:

a. daca procesul trateaza semnalul si acest lucru e posibil, se apeleaza o functie din acel proces

b. daca procesul nu trateaza semnalul este intotdeauna terminat

c. procesul poate desemna un alt proces scris pentru a trata semnalul

d. procesul poate sa ignore semnalul indiferent de caz

e. procesul trimite automat un semnal proceselor fiu

49. Diferenta dintre `execvp()` si `execl()` este:

a. `execvp` specifica o cale iar `execl` nu

b. `execvp` tine cont de date memorate intr-o anumita variabila de mediu

c. `execvp` lanseaza un proces pe cand `execl` nu

d. `execvp()` lanseaza un proces, pe cand `execl` executa un fisier de comenzi dat

e. `execvp()` este nerecomandat spre a fi folosit, spre deosebire de `execl()`

50. Un proces este:

a. un program sau o parte de program aflat(a) sub supervizarea sistemului de operare

b. o comanda interna

c. un program c care apeleaza `fork()`

d. un serviciu oferit de sistemul de operare programelor

e. o functie sistem care implementeaza serviciul de multitasking

51. open() este:

a. un apel sistem

b. o comanda care deschide un fisier

c. o functie de biblioteca

d. o functie aflata in interfata de programare UNIX

e. o comanda implementata in nucleul sistemului de operare

52. Comanda grep 'a.[os].\$ primind pe cate o linie sirurile de caractere de mai jos le va afisa la isesirea standard pe urmatoarele:

a. axpz

b. megafonul

c. rtatry

d. afon

e. megafon

53. Codul de mai jos va afisa la iesirea standard:

```
Int main(int argc, char **argv){ int pid1,pid2;
```

```
If(pid1=fork())<0){ printf("Error\n"); exit(1);}
```

```
Printf("B\n");
```

```
If(pid2=fork())<0){ printf("Error\n"); exit(2);}
```

```
Printf("A\n");
```

```
Return 0;
```

```
}
```

a. 3 caractere B

b. 2 caracter B

c. 2 caractere A

d. 5 caractere A

e. 1 caracter B

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

a. write(pfd[0],s, strlen(s))

b. write(pfd[0], &s, sizeof(s))

c. write(pfd[1],s, strlen(s))

d. write(pfd[1],s, sizeof(s))

e. write(pfd[1], &s, strlen(s))

15. Given the following code:

```
int rec(char *home){
    DIR* d;
    struct dirent* entry;
    struct stat buf;
    int count= 0;
    char path[100];

    if ((d= opendir(home)) == NULL){
        printf("Error opendir on %s\n", home);
        exit(1);
    }

    while ((entry = readdir(d)) != NULL){
        if( strcmp(entry->d_name, ".") == 0 || strcmp(entry->d_name, "..") == 0 ) continue;

        strcpy(path, home);
        strcat(path, "/");
        strcat(path, entry->d_name);

        if( stat( path, &buf) <0 ) {
            printf("Error stat on %s\n", path);
            exit(2);
        }

        if( S_ISDIR(buf, st_mode)
            count += rec(path);
        else{
            if(S_ISREG(buf, st_mode))
                count++;
        }

        closedir(d);

        return count;
    }
```

a) The regular files from dir_path are counted;
b) Execution error ("Error stat on..." because file is not found, e.g. (No such file or directory))
c) Those symbolic links from dir_path that refer to regular files are counted
d) Infinite loop
e) The subdirectories from dir_path are counted.

a) Si c)

56. x=10

Ce va afisa x\$x\$?

x10\$

57. Ce combinatie de comenzi va schimba directorul curent?

cd

58. Ce face functia signal()?

Specifica pentru procesul curent modul de tratare a unui semnal :P

59. Functie cu read() de citit dintr-un pipe

60. Cate procese vor fi create?

```
For(i=1;i<n;i++)
```

```
{fork();}
```

$2^n - 1$

61. Symlink path, bytesize ?

a. lstat

b. stat

c. readlnk

d. ln

62.The “wc -l” command counts:

a. Characters received at the standard input

b. Lines written to the standard output

c. Columns written to the standard output

d. Columns received at the standard input

e. Lines received at the standard input

63. var=3

```
Echo $var && { var="abc"; echo "$var";}
```

```
Echo $var
```

a. 3, abc

b. 3, 3

c. 3, abc, abc

d. 3, abc, 3

64. A process reads from multiple pipes and always has to process the first available data in any of the pipes. For this the programmer must use:

- a. sigaction()
- b. The SIGPIPE signal
- c. fread()
- d. select()
- e. signal()

65. The signal() function:

- a. Signals an error
- b. Handles a signal
- c. Sets the behavior for handling a signal
- d. Signals a process to end
- e. Sends a signal

66. Under what circumstances is the return value of read() function lower than its third parameter?

- a. When the last part of a file is being read
- b. In case of an error
- c. When the file size (in bytes) is lower than the requested byte count
- d. When data is read from a pipe and the processes that use the write end of the pipe no longer send data
- e. When there are more unread bytes in the file than the requested byte count

67. The standard out can be redirected to a file for which a valid descriptor fd exists, as follows:

- a. open(1, fd)

- b. `redirect(fd, 1)`
- c. `dup2(1, fd)`
- d. `dup2(fd, 1)`
- e. `redirect(1, fd)`

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

- a. Child process
- b. Daemon process
- c. Zombie process
- d. Parent process
- e. Orphan process

69. If a signal is sent to a process, then:

- a. The process may assign a process written specifically to handle the signal
- b. The process can ignore it, regardless of the signal
- c. If the process does not handle the signal, the process is always terminated
- d. The process automatically sends a signal to all child processes
- e. If the process handles the signal and it is possible to do this, then a function in that process is called

70. `var=23`

`Echo '`expr $var + 3`'`

`Echo "`expr $var + 3`"`

- a. 26, `'expr $var + 3`'`
- b. 26, 26
- c. ``expr $var + 3``, ``expr $var + 3`'`
- d. ``expr $var + 3``, 26

71. The correct call(s) - with valid arguments only!- to create a file to later write in it is

- a. `creat(file, O_WRONLY | O_CREAT | O_TRUNC)`
- b. `open(file, O_WRONLY | O_CREAT | O_TRUNC | S_IRUSR | S_IWUSR)`
- c. `creat(file, S_IRUSR | S_IWUSR)`
- d. `open(file, O_WRONLY | O_CREAT | O_TRUNC)`
- e. `open(file, O_WRONLY | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR)`

72. Examples of operating systems are:

- a. `MS-DOS`
- b. `UNIX`
- c. `Tizen`
- d. Apple
- e. Java

73. Select only the true statements

- a. The stack is shared among threads
- b. `The operating system offers an API`
- c. `A UNIX process may contain several threads`
- d. The data segment of a process is shared between all the threads in the system
- e. The operating system runs exclusively in the user space

74. The target of a double indirection entry from the specific data block list stored in the inode is

- a. Another inode
- b. `A block with pointers to other blocks`
- c. A data block of the file
- d. Double file blocks
- e. A symbolic link to the file

75. `ls` is, in UNIX and bash

- a. An external command
- b. An internal command
- c. A system call
- d. An internal function
- e. An external function

76. A process can always gain root privileges if:

- a. The real UID of the process was root in the past
- b. The effective UID was root in the past
- c. The real UID for the process is root
- d. The setuid flag of its configuration file is set
- e. Its saved set UID is root