

## subiecteSO\_eu.pdf

lesirea standard se poate redirecta catre capatul corespunzator al unui pipe p, create anterior cu apelul pipe() astfel:

Question **4**

Complete

Mark 0.00 out of 10.00

Flag question

lesirea standard se poate redirecta catre capatul corespunzator al unui pipe p, creat anterior cu apelul pipe(), astfel:

- ☒ a. `dup2(p[0], 1);` NO
- ☐ b. `open(1, p[1]);`
- ☐ c. `dup2(1, p[1]);`
- ☒ d. `dup2(p[1], 1);`
- ☐ e. `redirect(p[0], 1);`

Un blu-ray player care poate accesa servicii de streaming(Youtube, Netflix) are system de operare?

Question **3**

Complete

Mark 10.00 out of 10.00

Flag question

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

- ☐ a. nu, pentru ca e format din componente proprietare
- ☐ b. nu, pentru ca e deservit de un firmware
- ☐ c. da, pentru ca e un dispozitiv electronic
- ☒ d. da, pentru ca e un sistem de calcul cu procesor si dispozitive periferice
- ☐ e. nu, pentru ca este un dispozitiv electronic de uz general

Ce este /dev/null?

Question **6**

Complete

Mark 10.00 out of 10.00

Flag question

Ce este /dev/null ?

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

Ce functie sistem puteti folosi ca programator in UNIX pentru a sterge un fisier?

Functia creat() in UNIX:

Question 1

Complete

Mark 0.00 out of 10.00

Flag question

Ce functie sistem puteti folosi ca programator in UNIX pentru a sterge un fisier?

- ☐ a. remove
- ☒ b. unlink
- ☐ c. rmdir
- ☐ d. delete
- ☒ e. rm

NO

Question 2

Complete

Mark 10.00 out of 10.00

Flag question

Functia creat() in UNIX

- ☐ a. creeaza un proces
- ☒ b. creeaza un fisier
- ☐ c. creeaza un pipe
- ☐ d. creeaza un fir de executie
- ☐ e. creeaza o variabila de mediu

Cum se efectueaza correct (cu argumente corecte!) deschiderea unui fisier existent file pentru citire?

Question 7

Complete

Mark 10.00 out of 10.00

Flag question

Cum se efectueaza corect (cu argumente corecte!) deschiderea unui fisier existent file pentru citire?

- ☒ a. `open(file, O_RDONLY);`
- ☐ b. `open(file, O_RDONLY, S_IRUSR);`
- ☐ c. `open(file, O_WRONLY);`
- ☐ d. `open(file, O_WRONLY, S_IWUSR);`
- ☐ e. `open(file, O_RDONLY | S_IRUSR);`

Cate procese noi creeaza urmatorul cod:

Question **8**

Complete

Mark 10.00 out of 10.00

Flag question

Cate procese noi creeaza urmatorul cod:

```
int i; for(i=0; i<=2;i++) fork();
```

- ☐ a. 1
- ☐ b. 5
- ☐ c. 2
- ☐ d. 3
- ☒ e. 7

Daca un semnal e trimis unui process, atunci:

Question **9**

Complete

Mark 10.00 out of 10.00

Flag question

Daca un semnal e trimis unui proces, atunci:

- ☐ a. procesul poate delega tratarea semnalului unui alt proces destinat acestui scop
- ☐ b. daca procesul nu trateaza semnalul, procesul este intotdeauna terminat
- ☐ c. procesul poate sa il ignore, indiferent de semnal
- ☒ d. daca procesul trateaza semnalul si acest lucru e posibil, se apeleaza o functie din acel proces
- ☐ e. procesul trimite automat un semnal tuturor proceselor fiu

Un process poate intotdeauna sa primeasca privilegii de root, daca:

Question **10**

Complete

Mark 0.00 out of 10.00

Flag question

Un proces poate întotdeauna să primească privilegii de root dacă:

- NO
- ☒ a. effective UID al procesului a fost root în trecut
  - ☒ b. real UID al procesului este root
  - ☒ c. saved set UID al procesului este root
  - ☐ d. flag-ul setuid al fișierului său de configurare este setat
  - ☐ e. real UID al procesului a fost root în trecut

Ce se va afisa ca urmare a executiei urmatoarelor comenzi?

Question **12**  
Complete  
Mark 10.00 out of 10.00  
Flag question

Ce se va afisa ca urmare a executiei urmatoarelor comenzi?

```
x=10  
echo "x$x"
```

- ☐ a. x\$x\$
- ☐ b. mesaj de eroare
- ☐ c. 1010
- ☒ d. x10\$

Daca un shell script este apelat dupa cum urmeaza:

Question **13**  
Complete  
Mark 10.00 out of 10.00  
Flag question

Daca un shell script este apelat dupa cum urmeaza:

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

ce se va afisa ca urmare a executiei urmatoarei comenzi?

```
echo $0 $1 $2
```

- ☐ a. 1 2 3
- ☐ b. mesaj de eroare
- ☐ c. sh script.sh 1
- ☒ d. script.sh 1 2

Sub\_Stefi.pdf

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

Question **1**  
Complete  
Mark 0.00 out of 10.00  
Flag question

Numărul de blocuri de date ocupate de un fișier pe disc se poate afla:

- ☐ a. aflând adresa pe disc a ultimului bloc ocupat de fișier
- ☒ b. însumând informațiile din tabela de blocuri ocupate, aflată în nodul index
- ☐ c. citind cele trei intrări de îndirectare din tabela de blocuri
- ☐ d. parcurgând primele 10 intrări din tabela blocurilor ocupate
- ☒ e. citind o informație din nodul index și aplicând o formula care depinde și de dimensiunea blocului

Un programator C primește parametrii din linia de comandă în variabila `argv`. Dacă este compilat și apelat ca mai jos, care este `argv[2]`?

Question **2**

Complete

Mark 10.00 out of 10.00

Flag question

Un program C primește parametrii din linia de comandă în variabila `argv`. Dacă e compilat și apelat ca mai jos, care e `argv[2]`?

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

- ☐ a. 1 2
- ☒ b. 3 1
- ☐ c. 3
- ☐ d. 1
- ☐ e. 2

La terminare, un program returnează în mod obligatoriu către apelant

Question **3**

Complete

Mark 10.00 out of 10.00

Flag question

La terminare, un program returnează în mod obligatoriu către apelant

- ☐ a. o valoare întreagă la ieșirea standard
- ☒ b. un număr întreg
- ☐ c. un număr natural la ieșirea standard
- ☐ d. un sir de caractere
- ☐ e. un număr real la ieșirea standard

Un pipe trebuie creat în program

Question **5**

Complete

Mark 10.00 out of 10.00

Flag question

Un pipe trebuie creat în program

- ☐ a. numai în procesele care au creat deja un proces
- ☐ b. neapărat înainte de crearea tuturor proceselor fiu
- ☐ c. după ultimul `fork()` din program
- ☒ d. înainte de crearea proceselor care au nevoie de el
- ☐ e. imediat după crearea proceselor care îl folosesc

Ieșirea standard se poate redirecta către un fișier, pentru care există un descriptor valid `fd`, astfel:

Question **6**

Complete

Mark 10.00 out of 10.00

Flag question

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

☐ a. `redirect(fd, 1);`

☐ b. `redirect(1, fd);`

☐ c. `open(1, fd);`

☒ d. `dup2(fd, 1);`

☐ e. `dup2(1, fd);`

## O rutina de tratare poate

Question **8**

Complete

Mark 3.33 out of 10.00

Flag question

O rutina de tratare poate:

☐ a. modifica valoarea unei variabile declarate in main

☒ b. trimite un semnal

☐ c. sa se execute doar o singura data, indiferent de configurarea apelului `sigaction` si de cate ori este primit semnalul pe care il trateaza

☒ d. trata mai multe semnale diferite, facand uz de argumentul primit

☒ e. termina procesul curent

## Cate procese noi creeaza urmatorul cod

Question **10**

Complete

Mark 10.00 out of 10.00

Flag question

Cate procese noi creeaza urmatorul cod:

```
int i; for(i=0; i<2; i++) fork();
```

☒ a. 3

☐ b. 2

☐ c. 1

☐ d. 5

☐ e. 4

Cum se efectueaza correct (cu argument corect!) crearea unui fisier file in care se va scrie continut?

Question **11**

Complete

Mark 10.00 out of 10.00

Flag question

Cum se efectueaza corect (cu argumente corecte!) crearea unui fisier *file* in care se va scrie continut?

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

Alegeti varianta/variantele corecte de cod sursa dintre urmatoarele variantele, atunci cand se doreste scrierea unui string intr-un pipe

Question **12**

Complete

Mark 10.00 out of 10.00

Flag question

Alegeti varianta/variantele corecte de cod sursa dintre urmatoarele variante, atunci cand se doreste scrierea unui string intr-un pipe:

```
#define SIZE 20
int pfd[2];
char string_to_write[SIZE];
/*Initialize or compute string_to_write*/
```

- ☐ a. `write(pfd[1], string_to_write, SIZE);`
- ☐ b. `write(pfd[1], string_to_write, sizeof(string_to_write));`
- ☐ c. `write(pfd[0], string_to_write, strlen(string_to_write));`
- ☒ d. `write(pfd[1], string_to_write, strlen(string_to_write));`
- ☐ e. `write(pfd[0], string_to_write, sizeof(string_to_write));`

Alegeti varianta/variantele corecte de cod sursa dintre urmatoarele variantele, pentru a verifica daca programul a primit minim 2 argumente in linie de comanda

Question 13

Complete

Mark 0.00 out of 10.00

Flag question

Alegeti varianta/variantele corecte de cod sursa dintre urmatoarele variante, pentru a verifica daca programul a primit minim 2 argumente in linie de comanda.

```
int main (int argc, char* argv[]) { // alegeti varianta/variantele }
```

☐ a.

```
if (argc < 2) { perror("Apelare corecta: executabil argument1 argument2"); exit(1); }
```

☒ b.

```
if (argc < 3) { printf("Apelare corecta: %s argument1 argument2", argv[0]); exit(1); }
```

☐ c.

```
if (argc < 2) { printf("Apelare corecta: %s argument1 argument2", argv[0]); exit(1); }
```

NO ☒ d.

```
if (argc < 3) { perror("Apelare corecta: %s argument1 argument2", argv[0]); exit(1); }
```

NO ☒ e.

```
if (argc <= 2) { perror("Apelare corecta: %s argument1 argument2", argv[0]); exit(1); }
```

Sub\_Morariu.pdf

Exemple de sisteme de operare sunt

Question 1

Complete

Mark 6.67 out of 10.00

Flag question

Exemple de sisteme de operare sunt:

☐ a.

Java

☒ b.

MS-DOS

☒ c.

UNIX

☐ d.

Apple

☒ e.

Tizen

Sistemul de operare UNIX ofera, ca functionalitati specifice

Question 2

Complete

Mark 6.67 out of 10.00

Flag question

Sistemul de operare UNIX ofera, ca functionalitati specifice:

☐ a.

gestiunea bazelor de date

☒ b.

gestiunea proceselor

☒ c.

gestiunea memoriei

☐ d.

gestiunea site-urilor Web

☒ e.

interfata de programare

Un pipe functioneaza:



Question 3

Complete

Mark 10.00 out of 10.00

Flag question

Un pipe functioneaza:

- ☐ a. dupa regula primul venit, ultimul servit
- ☒ b. ca un canal de date unidirectional
- ☐ c. ca un canal de date bidirectional
- ☐ d. pe principiul LIFO
- ☒ e. pe principiul FIFO

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

Question 5

Complete

Mark 10.00 out of 10.00

Flag question

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. în execuție
- ☐ b. trimis in swapping
- ☒ c. blocat
- ☐ d. terminat
- ☐ e. pregatit de execuție

Nucleul unui system de operare bazat pe arhitectura microkernel



Question 8

Complete

Mark 0.00 out of 10.00

Flag question

Nucleul unui sistem de operare bazat pe arhitectura microkernel

-  a. e parte a unei arhitecturi flexibile, cu mentenanta usoara
- NO ☒ b. este format din mai multe servere independente care ofera servicii
- ???? ☐ c. are dezavantajul ca ocupa singur toata memoria
- ☐ d. este un concept pur teoretic, care nu a fost niciodata implementat tehnic
-  e. ofera servicii de baza, cum ar fi gestiunea proceselor

Daca un semnal e trimis unui process, atunci:

Question **9**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

Daca un semnal e trimis unui proces, atunci:

- ☐ a. procesul poate delega tratarea semnalului unui alt proces destinat acestui scop
- ☒ b. daca procesul trateaza semnalul si acest lucru e posibil, se apeleaza o functie din acel proces
- ☐ c. procesul trimite automat un semnal tuturor proceselor fiu
- ☐ d. daca procesul nu trateaza semnalul, procesul este intotdeauna terminat
- ☐ e. procesul poate sa il ignore, indiferent de semnal

Un process primește date din mai multe pipe-uri si trebuie sa proceseze imediat primele date disponibile indiferent de pipe-ul prin care ele sosesc. Pentru acest scop particular, programatorul trebuie sa foloseasca

Question **10**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

Un proces primește date din mai multe pipe-uri și trebuie să proceseze imediat primele date disponibile indiferent de pipe-ul prin care ele sosesc. Pentru acest scop particular, programatorul trebuie să folosească

- ☐ a. semnalul SIGPIPE
- ☐ b. sigaction()
- ☐ c. fread()
- ☐ d. signal()
- ☒ e. select()

Codul de mai jos va afisa la iesirea standard:

Question **13**

Complete

Mark 0.00 out of 10.00

🚩 Flag question

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;
}
```

NO ☒ a. 3 caractere B

☐ b. 1 caracter B

☒ c. 4 caractere B

☒ d. 2 caractere C

☐ e. 1 caracter C

Sub\_Sonia.pdf

Atunci cand un proces A apeleaza fork() pentru a crea un process B, procesul B este:

Question **1**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

Atunci cand un proces A apeleaza fork() pentru a crea un proces B, procesul B este:

- ☒ a. proces fiu
- ☐ b. proces părinte
- ☐ c. proces orfan
- ☐ d. proces zombie
- ☐ e. proces daemon

Un pipe functioneaza:

Question **2**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

Un pipe functioneaza:

- ☐ a. ca un canal de date bidirectional
- ☒ b. pe principiul FIFO
- ☐ c. dupa regula primul venit, ultimul servit
- ☒ d. ca un canal de date unidirectional
- ☐ e. pe principiul LIFO

Functia execlp()

Funcția `execvp()`

- ☐ a. executa o linie de program
- ☐ b. preia un program de pe disc si il executa intr-un proces nou
- NO ☒ c. executa o comanda UNIX si ii preia rezultatul
- ☒ d. executa un program de pe disc in cadrul procesului curent
- ☐ e. creeaza un proces si il executa

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

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

- ☒ a. da, pentru ca e un sistem de calcul cu procesor si dispozitive periferice
- ☐ b. nu, pentru ca este un dispozitiv electronic de uz general
- ☐ c. nu, pentru ca e deservit de un firmware
- ☐ d. da, pentru ca e un dispozitiv electronic
- ☐ e. nu, pentru ca e format din componente proprietare

Un process mosteneste de la procesul care l-a create:

Un proces mosteneste de la procesul care l-a creat:

- ☐ a. identificatorul de proces
- ☒ b. toate variabilele globale
- ☐ c. toate capetele de pipe-uri, indiferent daca au fost initializate inainte sau dupa creare
- ☒ d. fisierele deschise inainte de creare
- ☒ e. redirectarile de descriptori efectuate inainte de creare

eu.pdf

Ce functie system puteti folosi ca programator in UNIX pentru a sterge un fisier?

Question 2

Complete

Mark 10.00 out of 10.00

Flag question

Ce functie sistem puteti folosi ca programator in UNIX pentru a sterge un fisier?

- ☐ a. delete
- ☐ b. remove
- ☐ c. rm
- ☐ d. rmdir
- ☒ e. unlink

Java Virtual Machine este o masina virtuala:

Question 3

Complete

Mark 10.00 out of 10.00

Flag question

Java Virtual Machine este o masina virtuala

- ☐ a. specifica UNIX
- ☐ b. container
- ☐ c. implementata in interiorul nucleului sistemului de operare UNIX
- ☐ d. tip sistem
- ☒ e. tip proces

Pentru a afla spre ce cale indica o legatura simbolica si dimensiunea legaturii pe disc e nevoie de functii:

Question **6**

Complete

Mark 10.00 out of 10.00

Flag question

Pentru a afla spre ce cale indica o legatura simbolica si dimensiunea legaturii pe disc e nevoie de functiile:

- ☐ a. unlink
- ☐ b. stat
- ☒ c. readlink
- ☐ d. ln
- ☒ e. lstat

Marcati in foaia de raspuns numai afirmatiile adevarate de mai jos:

Question **10**

Complete

Mark 10.00 out of 10.00

Flag question

Marcati in foaia de raspuns numai afirmatiile adevarate de mai jos:

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

In UNIX, programele implementate de utilizator:

Question **11**

Complete

Mark 6.67 out of 10.00

Flag question

In UNIX, programele implementate de utilizator:

- ??? ☐ a. pot fi facute sa comunice intre ele, prin redirectari, indiferent de modul in care sunt implementate
- ☐ b. sunt obligate sa isi ia datele de la intrarea standard
- ☒ c. au posibilitatea de a redirecta iesirea standard
- ☒ d. pot fi formate din mai multe procese
- ☒ e. pot sa trimita date altor programe prin intermediul unor facilitati ale sistemului de operare

P2.pdf

Cand se recomanda folosirea unui buffer de dimensiunea unui bloc pentru optimizarea?

Question **5**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

Cand se recomanda folosirea unui buffer de dimensiunea unui bloc pentru optimizare?

- ☐ a. scriere in pipe
- ☒ b. citire din fisier
- ☒ c. scriere in fisier
- ☐ d. citirea nodului index
- ☐ e. citire din pipe

Intrarea de dubla indirectare aflata in tabelul din nodul index refera:

Question **6**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

Intrarea de dubla indirectare aflata in tabelul din nodul index refera:

- ☐ a. blocurile duble ale fișierului
- ☐ b. un alt nod index
- ☐ c. un bloc de date al fișierului
- ☒ d. un bloc conținând adrese de bloc
- ☐ e. o legătură simbolică spre fișier

O comanda externa este:

Question **7**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

O comanda externa este:

- ☐ a. o functie al carei cod nu se afla definit in programul curent
- ☒ b. un fisier executabil, aflat intr-un director a carui cale se afla in variabila de mediu PATH in mod implicit
- ☐ c. un shell script scris de utilizator
- ☐ d. un fisier executabil, aflat intr-un director a carui cale se afla in variabila de mediu TERM in mod implicit
- ☐ e. un apel sistem aflat in afara nucleului sistemului de operare

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



Question **8**  
Complete  
Mark 10.00 out of 10.00  
Flag question

Marcati in foaia de raspuns numai afirmatiile adevarate de mai jos:

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

In UNIX, un fir de executie poate:

Question **10**  
Complete  
Mark 10.00 out of 10.00  
Flag question

In UNIX, un fir de executie poate:

- ☐ a. defini un set de variabile globale accesibile doar lui insusi
- ☒ b. apela functii
- ☒ c. returna o valoare
- ☒ d. termina procesul curent
- ☐ e. accesa memoria oricarui alt fir de executie care ruleaza pe calculator

In UNIX, programele implementate de utilizator:

Question **11**  
Complete  
Mark 10.00 out of 10.00  
Flag question

In UNIX, programele implementate de utilizator:

- ☐ a. sunt obligate sa isi ia datele de la intrarea standard
- ☐ b. pot fi facute sa comunice intre ele, prin redirectari, indiferent de modul in care sunt implementate
- ☒ c. pot sa trimita date altor programe prin intermediul unor facilitati ale sistemului de operare
- ☒ d. au posibilitatea de a redirecta iesirea standard
- ☒ e. pot fi formate din mai multe procese

Comanda grep primind pe cate o linie sirurile de mai jos, le va afisa la iesirea standard pe urmatoarele:

Question **13**

Complete

Mark 10.00 out of 10.00

Flag question

Comanda

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

primind pe cate o linie sirurile de mai jos, le va afisa la iesirea standard pe urmatoarele:

- ☐ a. s1.text
- ☒ b. os1.text
- ☐ c. conf.t
- ☐ d. s1.ttm
- ☐ e. config.tmp

p3\_2021.pdf

ls este, in UNIX si bash

Question **2**

Complete

Mark 10.00 out of 10.00

Flag question

ls este, in UNIX si bash

- ☒ a. o comanda externa
- ☐ b. o comanda interna
- ☐ c. o functie interna
- ☐ d. o functie externa
- ☐ e. o functie sistem

Un process este

Question **4**

Complete

Mark 10.00 out of 10.00

🚩 Flag question

Un proces este

- ☐ a. un program C care apeleaza fork()
- ☐ b. un serviciu oferit de sistemul de operare programelor
- ☐ c. o comanda interna
- ☐ d. o functie sistem care implementeaza serviciul de multitasking
- ☒ e. un program sau o parte de program aflat(a) sub supervizarea sistemului de operare

Planificarea la executie a proceselor este:

Question **6**

Complete

Mark 10.00 out of 10.00

🚩 Remove flag

Planificarea la executie a proceselor este

- ☐ a. o comanda care planifica executia programelor la diferite momente de timp
- ☐ b. un apel sistem
- ☒ c. o functionalitate implementata de sistemul de operare
- ☐ d. o facilitate existenta in toate interpretoarele de comenzi moderne
- ☐ e. un algoritm de gestiune a drepturilor de executie pentru programe

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

Question **10**

Complete

Mark 10.00 out of 10.00

🚩 Remove flag

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

- ☒ a. un alt proces poate citi doar jumatate din structura, in anumite conditii
- ☐ b. apare intotdeauna eroare
- ☒ c. poate exista un proces care poate citi toata structura
- ☐ d. un proces e obligat sa citeasca toata structura, altfel primeste cod de eroare
- ☐ e. orice proces din sistem poate citi structura

In UNIX, programele implementate de utilizator:

Question **11**

Complete

Mark 10.00 out of 10.00

Flag question

In UNIX, programele implementate de utilizator:

- ☒ a. pot fi formate din mai multe procese
- ☒ b. au posibilitatea de a redirecta iesirea standard
- ☐ c. sunt obligate sa isi ia datele de la intrarea standard
- ☒ d. pot sa trimita date altor programe prin intermediul unor facilitati ale sistemului de operare
- ☐ e. pot fi facute sa comunice intre ele, prin redirectari, indiferent de modul in care sunt implementate

so\_p2.pdf

Un fir de executie in UNIX este

Question **1**

Complete

Mark 10.00 out of 10.00

Flag question

Un fir de executie in UNIX este

- ☐ a. o functie definita intr-un shell script
- ☐ b. o functie care implementeaza un algoritm de executie
- ☐ c. un proces terminat
- ☐ d. un set de comenzi executat in bash
- ☒ e. o executie secventiala in cadrul unui proces

Comanda grep

Question **2**

Complete

Mark 10.00 out of 10.00

Flag question

Comanda grep

- ☐ a. returneaza numarul de tipare sterse
- ☐ b. numara tiparele gasite intr-un text
- ☐ c. returneaza numarul de tipare gasite
- ☐ d. sterge tiparele gasite intr-un text
- ☒ e. cauta potriviri pentru un tipar dat

Legaturile fixe sunt

Question **3**

Complete

Mark 10.00 out of 10.00

Flag question

Legăturile fixe sunt:

- ☐ a. nume alternative pentru același fișier, aflate întotdeauna în același director
- ☐ b. fișiere individuale care referă alte fișiere
- ☒ c. intrări în director diferite care referă același nod index
- ☐ d. fișiere speciale
- ☐ e. fișiere individuale care referă directoare fixe

O comanda externa:

Question **4**

Complete

Mark 10.00 out of 10.00

Flag question

O comanda externa este:

- ☐ a. o funcție al cărei cod nu se afla definit în programul curent
- ☐ b. un fișier executabil, aflat într-un director a cărui cale se afla în variabila de mediu TERM în mod implicit
- ☐ c. un shell script scris de utilizator
- ☐ d. un apel sistem aflat în afara nucleului sistemului de operare
- ☒ e. un fișier executabil, aflat într-un director a cărui cale se afla în variabila de mediu PATH în mod implicit

Un fișier poate fi făcut disponibil pentru două procese astfel

Question **8**

Complete

Mark 10.00 out of 10.00

Flag question

Un fișier poate fi făcut disponibil pentru două procese astfel:

- ☒ a. un proces trimite către celălalt numele fișierului
- ☐ b. procesul care deschide fișierul trimite descriptorul de fișier obținut la `open()` celuiălalt proces
- ☐ c. un proces creează celălalt proces, iar acesta din urmă creează fișierul
- ☐ d. se folosește funcția `fdopen` care trimite un identificator tip `FILE *` al fișierului
- ☒ e. un proces creează fișierul și apoi creează celălalt proces

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

Flagurile `SETUID` și `SETGID` se referă la:




Question **9**

Complete

Mark 6.67 out of 10.00

 Remove flag

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

-  a. cand se citeste ultimul set de date dintr-un fisier
- ☐ b. cand in fisier mai exista necititi un numar mai mare de bytes decat cel cerut
-  c. cand fisierul citit e mai mic, in bytes, decat numarul de octeti cerut
-  d. in caz de eroare
- ☐ e. cand se citesc date dintr-un pipe si procesele conectate la capatul de scriere nu mai furnizeaza date


Question **10**

Complete

Mark 0.00 out of 10.00

 Remove flag

Flag-urile SETUID și SETGID se referă la

- ☐ a. privilegiile de execuție din trecut ale unui proces
- ☐ b. privilegiile salvate ale unui proces aflat în execuție
- ☒ c. privilegiile de execuție actuale ale procesului
- ☐ d. drepturile de modificare a unui fișier
-  e. fișiere aflate în sistemul de fișiere

# SUBIECTE OS – TOT CE AM

## 1 DOC-UL INTREB

---

1. Marcati in foaia de raspuns numai afirmatiile adevarate de mai jos:

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

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

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

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

- a) Un numar intreg

4. Planificarea la executie a proceselor este:

- a) Un apel sistem

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

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

- a) dup2(fd,1)

7. O variabila de mediu bash

a) e de tipul sir de caractere

#### **8.Comanda wc -l numara**

a) liniile primite la intrarea standard

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

a) Parcurgând primele 10 intrari din tabela blocurilor ocupate

b) citind cele trei intrare de indirectare din tabela de blocuri

#### **10./dev/null este**

a) Un fisier special spre care orice date sunt scrise se pierd

#### **11.Nucleul unui sistem de operare bazat pe arhitectura microkernel**

a) Oferă servicii de baza, cum ar fi gestiunea proceselor

b) E parte a unei arhitecturi flexibile, cu mentenanța ușoară

c) Este format din mai multe servere independente care oferă servicii

#### **12.bash este:**

a) un shell

b) un interpretor de comenzi

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

a) da, pentru ca e un sistem de calcul cu processor si dispozitive periferice

#### **14.Un fisier poate fi facut disponibil pentru doua procese astfel:**

a) un proces creeaza fisierul si apoi il creeaza pe celalalt proces

b) ambele procese cunosc descriptorul de tip int al fisierului

#### **15.daca un proces trimite intr-un pipe o structura de date**

a) orice proces din sistem poate citi structura



b) un proces e obligat sa citeasca toata structura, altfel primeste cod de eroare

**16. fork() este:**

a) un apel care creeaza un proces

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

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

a) activeaza dreptul de read pentru proprietar

b) activeaza dreptul de executie pentru grup

**18. Java Virtual Machine este o masina virtuala**

b) tip proces

**19. informatiile despre drepturile de acces la un fisier se găsesc:**

a) in nodul index asociat acestuia

b) in sistemul de operare

**20. Un proces A creează un nou proces B, dupa care procesul A apeleaza cu succes functia execlp() pentru a porni un program P. Care vor fi consecintele?**

a) Programul nu va mai putea prelua explicit valoarea returnata la terminare de B

b) Procesul care ruleaza P va avea același PID ca A

**21. Functia kill()**

a) trimite un semnal

b) termina intodeauna un proces, in mod brusc

c) distruge fara drept de apel un proces

d) capteaza un semnal

e) trimite semnalul cu nr 9 catre comanda curenta

**22. Functia alarm()**

a) programeaza trimiterea unui semnal anume in viitor

**23. In UNIX, un fir de executie poate, daca este programat in acest scop**

- a) termina procesul curent
- b) apela functii
- c) returna o valoare
- d) semnaleaza o situatie de eroare aparuta in executie
- e) define un set de variabile globale accesibile doar lui insusi

**24. Legaturile fixe sunt:**

- a) intrari in director diferite care refera acelasi nod index

**25. Un programator doreste sa transmita un sir de dimensiune variabila mai micat 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**

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

**25. 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;
}
```

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

26. Un program C primește parametrii din linia de comandă în variabila argv. Dacă e compilat și apelat ca mai jos, care e argv[2]? ./program "1 2" "3 1" 2 3 '1' 2 3

- a) 1 -> argv[1]
- b) 2 -> argv[2]
- c) 1 2
- d) 3 1
- e) 3 -> argv[3]

27. Un router care poate fi administrat folosind un browser Web are sist de operare?

- a) da, pt ca e un system de calcul cu processor si dispozitive periferice

28. Linia urmatoare:

echo "cat f \ | wc -l" > g

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

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

- a) conf.t
- b) os1.text
- c) config.tmp
- d) s1.ttm
- e) s1.text

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

- a) axpz
- b) megafonul
- c) rtatry
- d) afon
- e) megafon

**31. Codul de mai jos va afisa la stdout:**

```
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\n");
    if((pid2 = fork())<0){
        printf("error\n");
        exit(2);
    }
    printf("A\n");
    return 0;
}
```

- a) 3 caract B
- b) 2 caract B**
- c) 2 caract A
- d) 5 caract A**
- e) 1 caract B

**32. Informatii despre nr de legaturi spre un fisier se gasesc:**

- a) in directorul current
- b) in directorul parinte
- c) in nodul index**
- d) in intrarea in director
- e) in campul st\_mode

**33. Java Virtual Machine este o masina virtuala:**

- a) tip process**
- b) tip system
- c) container
- d) implementata in interiorul nucleului sistemului de operare UNIX
- e) specifica UNIX

**34. Codul unei functii de biblioteca se afla:**

- a) in nucleul sistemului
- b) intr un header C ??**
- c) intr un fisier binary
- d) in partitia de swap
- e) in varful stivei ?

**35. Daca un process trimite cu success date unui alt process printr un pipe, inseamna in mod necesar ca:**

- a) procesele sunt fiii aceluiasi process distinct
- b) procesele au mostenit capetele necesare ale pipe ului
- c) exista un process tert care le este stramos comun celor 2 procese
- d) procesele au inchis capete nefolosite ale pipe ului
- e) procesele nu au inchis capetele utile ale pipe ului la care au acces

**36. Daca un semnal e trimis unui process, atunci:**

- a) daca procesul trateaza semnalul si acest lucru e posibil, se apeleaza o functie din acel process
- b) daca procesul nu trateaza semnalul, este intotdeauna terminat
- c) procesul poate desemna un proces scris special pt a trata semnalul
- d) procesul poate sa ignore semnalul, indiferent de caz
- e) procesul trimite automat un semnal tuturor proceselor fiu

**37. Diferenta dintre execlp() si execl este:**

- a) execlp() specifica o cale, dar execl nu
- b) execlp() tine cont de date memorate intr o anumita variabila de mediu, spre deosebire de execl care nu fol acea variabila
- c) execlp() lanseaza un process, pe cand execl nu
- d) execlp() lanseaza un process, pe cand execl() executa un fisier de comenzi dat
- e) execlp()este nerecomandat a fi folosit , spre deosebire de execl()

**38. Un process este:**

- a) un prog sau o parte din program aflat(a) sub supervizarea sist de operare
- b) o comanda interna
- c) un program C care apeleaza fork()
- d) un serviciu oferit de sist de op programelor
- e) o func system care implementeaza serviciul de multitasking

**39. open() este:**

- a) un apel system
- b) o comanda care deschide un fisier
- c) o functie de biblioteca
- d) o func aflata in interfata de programare
- e) o comanda care e implementata in nucleul sist de operare

## 2 DOC-UL 447

---

Subiecte OS

1. Marcati afirmatiile adevarate de mai jos:

- a) Sistemul de operare are acces la spatiul nucleu(kernel space)
- b) Memoria unui fir de executie este partajata intre toate procesele din acel fir
- c) Un proces UNIX poate fi format din mai multe procese
- d) Sistemul de operare ofera o interfata de programare pentru aplicatii
- e) Stiva e privata fiecarui proces

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

- a) Cand se citeste ultimul set de date dintr-un fisier
- b) Cand se citesc date dintr-un pipe si procesele conectate la capatul de scriere nu mai furnizeaza date
- c) In caz de eroare
- d) Cand in fisier mai exista necititi un numar mai mare de bytes decat cel cerut
- e) Cand fisierul citit e mai mic, in bytes, decat numarul de octeti ceruti

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

- a) Un numar intreg
- b) O valoare intreaga la iesirea standard
- c) Un numar natural la iesirea standard
- d) Un sir de caractere
- e) Un numar real la iesirea standard

4. Planificarea la executie a programelor este:

- a) Un apel sistem
- b) O comanda care planifica executia programelor la diferite momente de timp

c) O funcționalitate oferită de sistemul de operare

d) O facilitate existentă în toate interpretoarele de comenzi moderne

e) O facilitate oferită de toate shell-urile din UNIX

f) Un algoritm de gestiune a drepturilor de execuție pentru programe

5. Un proces aflat în execuție care apelează `read()` pe capatul de citire dintr-un pipe în care nu sunt date va fi trecut în starea:

a) blocat

b) pregătit de execuție

c) în execuție

d) trimis în swapping

e) terminat

6. Ieșirea standard se poate redirecta către un fișier pentru care există 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 variabilă de mediu `bash`:

a) e de tipul șir de caractere

b) este un număr întreg

c) este un număr natural

d) este un număr real

e) poate avea orice tip

8. Comanda `wc -l` numără:

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 creeaza 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 caluclatoare
- 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

Poate fi si database...!!!

..... 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 general
- 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 fii

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, scirul 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

77. A child process inherits from its parent:

- a. all the values on the stack as they were before the process was created
- b. the process identifier
- c. all global variables
- d. all file descriptors redirections
- e. all open files

### 3 DOC-UL GRILE ȘO

---

1. Java Virtual Machine este o masina virtuala:

- a. Tip sistem
- b. Tip proces
- c. Container
- d. Implementata in interiorul nucleului sistemului de operare UNIX
- e. Specifica UNIX

2. Informatiile despre drepturile de acces la un fisier 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
3. Un proces A creeaza un nou proces B dupa care procesul A apeleaza cu succes functia `execlp()` 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 (nu prea inteleg ce scrie, sper ca e corect)
  - c. Procesul care ruleaza P va avea acelasi PID ca A
  - d. Procesul care ruleaza P va avea alt PID decat A
  - e. Se termina automt pentru ca nu mai are parinte
4. Functia `kill()`
- a. Termina intotdeauna un proces, in mod brusc
  - b. Transmite un semnal
  - c. Distruge fara drept de apel un proces
  - d. Capteaza (maybe) un semnal
  - e. Trimite semnalul cu numarul 9 catre comanda curenta (cred ca asta scrie)
5. Functia `alarm()`
- a. Programeaza trimiterea unui semnal anume in viitor
  - b. Semnaleaza o situatie de eroare aparuta in executie
  - c. Ruleaza o alarma declansata
  - d. Informeaza un proces despre aparitia unei erori la alt proces
  - e. Anunta procesele ca trebuie sa se termine imediat
  - f. Reseteaza orice alarma setata anterior in acelasi process
6. 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 insuis
  - d. Apela functii
  - e. Memora o valoare
  - f. Accesa si modifica valorile variabilelor globale ale procesului in care fiul a fost creat
  - g. Accesa variabile declarate pe stiva in functiile asociate altor fire de executie
  - h. Returna o valoare
  - i. Exista in afara unui process
7. Legaturile fixe sunt:
- a. Nu vad..... Maybe intrari la director diferite care refera acelasi nod index
  - b. Nume alternative pentru acelasi ...., aflate intotdeauna intr-un acelasi director
  - c. Fisierse individuale care referera alte fisiere

- d. Fisiere individuale care refera directoare fixe
  - e. Fisierele speciale
8. Functia fork() creeaza
- a. Un program diferit de programul parinte, dar initial identic cu aceasta
  - 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
9. Un programator doreste sa traseze 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:
- 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));
10. Un router care poate fi administrat folosind un site Web are sistem de operare?
- a. Da, pentru ca e un dispozitiv electronic
  - b. Da, pentru ca e un sistem de calcul cu procesor si dispozitive periferice
  - c. Nu, pentru ca e deservit de firmware
  - d. Nu, pentru ca e un dispozitiv electronic de uz general
  - e. Nu, pentru ca e format din componente proprietare
11. Informatiile despre numarul de legaturi spre un fisier se gasesc:
- a. In directorul curent
  - b. In directorul parinte
  - c. In nodul index
  - d. In intrarea in director
  - e. In campul st\_mode
12. . Un program C primește parametrii din linia de comandă în variabila argv. Dacă e compilat și apelat ca mai jos, care e argv[2]?
- /program "1 2" "3 1" 2 3 '1' 2 3
- a. 3
  - b. 2
  - c. 1 2
  - d. 3 1

e. 3

13. Comanda `grep -E "^[st].+\.t[enpxt]*$"` primind pe cate o linie sirurile de mai jos va afisa la iesirea standard pe urmatoarele:

- a. Conf
- b. St.text
- c. Config.tmp
- d. St.dm
- e. pst.txt (chiar nu am vazut grilele alea, nu stiu daca am scris ce trebuie, too blurry)

14. 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; }
```

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

15. Marcati afirmatiile corecte:

- a. Sistemul de operare are acces la spatiul nucleu (kernel space)
- b. Memoria de date a unui fir de executie este partajata intre toate procesele din acel fir
- c. Un proces UNIX poate fi format din mai multe procese
- d. Sistemul de operare ofera o interfata de programare pentru aplicatii
- e. Stiva e privata fiecarui process

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

- a. Cand se citeste ultimul set de date dintr-un fisier
- b. Cand se citesc date dintr-un pipe si procesele conectate la capatul de scriere nu mai furnizeaza date
- c. In caz de eroare
- d. Cand in fisier mai exista necititi un numar mai mare de bytes decat cel cerut
- e. Cand fisierul citit e mai mic, in bytes, decat numarul de octeti ceruti

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

- a. Un numar intreg
- b. O valoare intreaga la iesirea standard
- c. Un numar natural la iesirea standard
- d. Un sir de caractere
- e. Un numar real la iesirea standard

18. Planificarea la executie a proceselor este:

- a. Un apel sistem
- b. O comanda care planifica executia programelor la diferite momente de timp
- c. O functionalitate oferita 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

19. 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

20. 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(1,fd)
- d. Redirect(fd,1)
- e. Dup2(1,fd)

21. 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

22. 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

23. 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 trei intrari de indirectare din tabela de blocuri
- d. Insumand informatiile din tabela de blocuri ocupate, aflate in nodul index
- e. Afland adresa pe disc a ultimului bloc ocupat de fisier

24. /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

25. Nucleul unui sistem de operare bazat pe arhitectura microkernel

- a. Este format din mai multe servere independente care 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, care nu a fost niciodata implementat

26. Bash este

- a. Un shell
- b. Un shell script
- c. Un apel sistem
- d. Un interpretor de comenzi
- e. O functie de biblioteca

27. Un blu-ray player care poate accesa servicii de streaming (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 este un dispozitiv electronic de uz general
- d. Da, pentru ca este un sistem de calcul cu procesor si dispozitive periferice
- e. Nu, pentru ca e format din componente proprietare

28. Un fisier poate fi facut disponibil pentru doua procese astfel:

- a. Procesul care deschide fisierul trimite descriptorul de fisier obtinut la open() celui alt proces
- b. Un proces trimite catre celalt numele fisierului
- c. Un proces creeaza fisierul si apoi il creeaza pe celalt proces
- d. Se foloseste de functia fdopen care trimite un identificator tip \*FILE al fisierului
- e. Ambele procese cunosc descriptorul de tip int al fisierului

29. Daca un pproces 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

30. Marcati afirmatiile adevarate:

- a. Sistemul de operare ruleaza exclusiv in spatiul utilizatorului (user space)
- b. Memoria de date a unui proces este partajata intre toate firele de executie intr-un 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

31. Fork() este

- a. O comanda care creeaza doua procese
- b. O functie care creeaza doua 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

32. 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 modul index si aplicand o formula care depinde si de dimensiunea blocului
- c. Citind cele trei intrari de indirectare din tabela de blocuri
- d. Insumand informatiile in tabela de blocuri ocupate, aflata in nodul index
- e. Afland adresa pe disc a ultimului bloc ocupat de fisier

33. 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

34. 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:

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

35. A process can always gain root privileges if: (Select one or more)

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

36. What system call can a programmer use in UNIX to delete a file?(Select one or more)

- a. `Unlink`
- b. `Rmdir`
- c. `Remove`
- d. `Delete`
- e. `Rm`

37. Information on access rights to a file reside in (Select one or more)

- a. The inode of the file
- b. The symbolic link
- c. The file
- d. The directory that contains the file
- e. The operating system

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

```
# define BLOCK_SIZE 4096
```

```
char buffer [BLOCK_SIZE];
```

```
Int fd =open(file_name, O_RDONLY);
```

- a. 

```
while ((n= read(fd, buffer, BLOCK_SIZE)) > 0)
    for(int i = 0; i < n; i++) {...}
```
- b. 

```
while ( read( fd, buffer, BLOCK_SIZE) > 0)
    for(int i = 0; i < strlen(buffer); i++ ) {...}
```



- c. 

```
while ((n= read(fd, buffer, BLOCK_SIZE)) > 0) {  
    buffer [n]= '\0' ;  
    for( int i = 0; i < strlen(buffer); i++) {...}  
}
```
- d. 

```
while (read(fd, buffer, BLOCK_SIZE) > 0)  
    for(int i = 0; buffer[i]; i++) {...}
```
- e. 

```
while((n = read(fd, buffer, BLOCK_SIZE)) > 0)  
    for(int i = 0; i < sizeof(buffer); i++) {...}
```

39. A process reads from multiple pipes and always has to process the first available data in any of the pipes.

For this specific purpose, the programmer must use

Select one or more:

- a. **Select()**
- b. Signal()
- c. Fread()
- d. Sigaction()
- e. The SIGPIPE signal

40. open() is

Select one or more:

- a. **a command implemented in the operating system kernel**
- b. a command that opens a file
- c. **system call**
- d. **a library function**
- e. **a function provided by the UNIX API**

41. The grep command

- a. Counts the patterns in a text
- b. Removes the patterns in a text
- c. **Finds matches for a given pattern**
- d. Returns the number of matching patterns
- e. Returns the number of removed patterns

42. The kernel of an operating system based on the microkernel architecture

- a. **Offers basic services, such as process management**
- b. Has the downside that it fills the entire memory on its own
- c. **Is part of a flexible architecture, with easy maintenance**
- d. **Is made of several independent servers that offer services**
- e. Is a purely theoretical concept that has never been implemented in practice

43. What characters (and how many times) will the following code display at the standard output?

```
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;
}
```

- a. "B" twice
- b. "B" once
- c. "A" five times
- d. "A" twice

44. A process is

- a. An internal command
- b. A program or part of a program under the supervision of the OS
- c. A system function that implements the multitasking service
- d. A C program that calls fork()
- e. A service offered by the operating system to the programs

45. When having one process A that creates another process B by calling fork(), after process A ends, process B is

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

46. Is it recommended to use block-sized buffers for optimization when

- a. Reading from a file
- b. Writing to a file
- c. Writing to a pipe
- d. Reading the index node
- e. reading from a pipe

---

incepe GX9

47. Codul unei functii de biblioteca se afla:

- a. In nucleul sistemului de operare
- b. Intr-un header
- c. Intr-un fisier binar
- d. In partitia de swap
- e. In varful stivei

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

- a. Procesul A nu va mai putea prelua explicit valoarea returnata la terminal de B
- b. Procesul A va fi terminat complet si un nou process va rula un program de pe disc
- 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

49. Informatia care specifica numarul de legaturi spre un fisier se gaseste:

- a. In directorul curent
- b. In directorul parinte
- c. In nodul index
- d. In intrarea in director
- e. In campul `st_mode`

50. Diferenta intre `execlp()` si `execl` este:

- a. `Execlp()` specifica o cale, dar `execl` nu
- b. Tine cont de date memorate intr-o anumita variabila de mediu, spre deosebire de `execl` care nu foloseste acea variabila
- c. `Execlp()` lanseaza un process, pe cand `execl` nu
- d. `Execlp()` lanseaza un proces, pe cand `execl()` executa un fisier de comenzi dat
- e. `Execlp()` este nerecomandat a fi folosit, spre deosebire de `execl()`

51. Un process 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

52. Open() este

- a. Un apel sistem
- b. O comanda care deschide un fisier
- c. O functie de biblioteca
- d. Functie aflata in interfata de programare oferita de UNIX
- e. O comanda care este implementata in nucleul sistemului de operare

53. Daca un semnal e trimis unui process, atunci:

- a. Daca procesul trateaza semnalul si acest lucru este posibil, se apeleaza o functie din acel process
- b. Daca procesul nu trateaza semnalul, procesul este intotdeauna terminat
- c. Procesul poate desemna un process scris special pentru a trata semnalul
- d. Procesul poate sa il ignore, indiferent de semnal
- e. Procesul trimite automat un semnal tuturor proceselor fiu

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

- a. Da, pentru ca e un dispozitiv electronic
- b. Da, pentru ca e un sistem de calcul cu processor si dispozitive periferice
- c. Nu, pentru ca e deservit de un firmware
- d. Nu, pentru ca e un dispozitiv electronic de uz general
- e. Nu, pentru ca e format din componente proprietare

55. Linia urmatoare : echo "cat f \ | wc -l " > g

- a. Scribe intr-un fisier dimensiunea(in octeti) unui alt fisier
- b. Afiseaza pe ecran dimensiunea unui fisier
- c. Compara numarul de linii dintr-un fisier cu o variabila g
- d. Scribe intr-un fisier numarul de linii din alt fisier
- e. Scribe intr-un fisier sirul de caractere cat f | wc -l

56. Legaturile fixe sunt:

- a. Intrari in 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

57. Planificarea la executie a proceselor este

- a. Un apel sistem

- b. O comanda care planifica executia programelor la diferite momente de timp
- c. O functionalitate implementata de sistemul de operare**
- d. O facilitate existenta in toate interpretoarele de comenzi moderne
- e. Un algoritm de gestiune a drepturilor

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

`./program "1 2" "3 1" "2 3" '1' 23`

- a. 1
- b. 2
- c. 1 2
- d. 3 1
- e. 3

E 2 3

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

- a. Conf.t
- b. Osl.text**
- c. Config.tmp
- d. Sl.tm
- e. S1.text

60. 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;
}
```

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

## 4 DOC-UL ANSWERS

---

### Answers

D60-B

1. Nucleul unui sistem de operare bazat pe arhitectura microkernel  
A?, bd
2. Bash este  
ad
3. Un blu-ray player care poate accesa servicii de streaming (eg. Netflix) are system de operare?  
d
4. Un fisier poate fi facut disponibil pentru doua procese astfel  
ce
5. Daca un proces trimite intr-un pipe o structura de date  
c
6. Afirmatii adevarate  
bcd
7. Fork() este  
ce
8. Nr de blocuri de date ocupate de un fisier pe disc se poate afla  
ac
9. Comanda chmod 512 x va avea urmatorul effect pentru fisierul x  
ad