

# exam questions

---

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 `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 (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 automat 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 proces

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

- ☒ a. Termina procesul curent (prin `exit()`)
- ☐ b. Accesa memoria oricarui alt fir de executie care ruleaza pe calculator (nu pe PC)
- ☐ 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 firul a fost creat
- ☐ g. Accesa variabile declarate pe stiva in functiile asociate altor fire de executie (stivele sunt independente)
- ☒ h. Returna o valoare
- ☐ i. Exista in afara unui proces

7. Legaturile fixe sunt:

- ☒ a. Intrari la director diferite care refera acelasi nod index
- ☐ b. Nume alternative pentru acelasi ...., aflate intotdeauna intr-un acelasi director
- ☐ c. Fisisere individuale care referera alte fisiere ???
- ☐ d. Fisisere individuale care refera directoare fixe
- ☐ e. Fisiererele speciale

8. Functia `fork()` creeaza

- ☒ a. Un program diferit de programul parinte, dar initial identic cu cacesta (Procesele copiaza zona de date, stiva si codul parintelui)
- ☐ b. Un fiu al programului curent incarcat de pe disc (asta e `exec()`)
- ☐ c. Un nou proces cu acelasi identificator ca procesul curent
- ☒ d. Un proces nou
- ☐ e. Un proces parinte

9. Un programator doreste sa trasnmita un sir de dimensiune variabila mai mica decar 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 `pfid` 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 varaibila 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 (./program e argv[0])
- ☐ e. 3

13. Comanda `grep -E "^.[st].+\.t[enpxt]*$"` (nota rezolvatorului: incepe cu orice caracter + 1 carcter din [st] + 1 sau mai multe orice caracter + . + t + se termina cu 0 sau mai multe caractere din [enpxt]) primind pe cate o linie sirurile de mai le va afisa la iesirea standard pe urmatoarele:

- ☐ a. Conf
- ☐ b. St.text
- ☐ c. Config.tmp
- ☐ d. [St.dm](#)

☒ e. pst.txt

14. Codul de mai jos va afișa la ieșirea standard (alegeți 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((pid1-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. 3caractere 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 proces

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 (se poate lol)
- ☒ b. Cand se citesc date dintr-un pipe si procesele conectate la capatul de scriere nu mai furnizeaza date
- ☒ c. In caz de eroare (-1 < num\_bytes right? riiiiight?)
- ☐ 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 ??? (comanda?)
- ☒ 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 ???(maybe this)
- ☐ 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 micokernel

- ☒ 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 (ocupa multa, nu toata i guess)
- ☒ d. E parte a unei arhitecturi flexibile, cu mentenanta usoara
- ☐ e. Este un concept pur teoretic, care nu a fost niciodata implemenat tehnic

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 celalalt 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 (program?)
- ☒ 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. Comanda chmod 512 (101 001 010 user group other rwx rwx rwx) 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. creat(file, S\_IRUSR | S\_IWUSR); (nu create)
- ☐ 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\_WRONLY | O\_CREAT |O\_TRUNC, S\_IRUSR | S\_IWUSR );
- ☐ e. creat(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 aet 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 (system call)
- ☐ b. Rmdir
- ☐ c. Remove (library call)
- ☐ d. Delete
- ☒ e. rm (rm uses unlink) ???

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++ ) {...} //might not have /0
```

☐ c.

```
while ((n= read(fd, buffer, BLOCK_SIZE)) > 0) {
    buffer [n]= '\0' ;
    for( int i = 0; i < strlen(buffer); i++) {...} //might overwrite the last char
}
```

☐ d.

```
while (read(fd, buffer, BLOCK_SIZE) > 0)
    for(int i = 0; buffer[i]; i++) {...} //might encounter \0 in the middle of the word
```

☐ e.

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

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. 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\n"); // original nu avea \n
    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

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

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() ???

<https://stackoverflow.com/questions/47193293/execlp-vs-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 specialpentru a trata semnalul
- ☐ d. Procesul poate sa il ignore, indiferent de semnal
- ☐ e. Procesul trimite automat un semnal tuturor proceselor fii

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

- ☐ a. Scrie 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. Scrie intr-un fisier numarul de linii din alt fiser
- ☐ e. Scrie intr-un fisier sirul de caractere cat f | wc -l