

CTRL + F intrebarile (daca aveti ceva ce nu e pe docs, adaugati aici TEXT)

GRESIT

CORECT

POATE

Dați și voi un search înainte să repetăți întrebarea.

Conform convenției de apel CDECL, o valoare de return pe 32 de biți se pune:

Conform convenției de apel CDECL, o valoare de return pe 32 de biți se pune:

- a. în registrul ESP
- b. pe stivă
- c. în registrul EIP
- d. în registrul EAX

[Clear my choice](#)



d) în registrul EAX

Este un utilitar folosit în detectarea abuzării bufferelor:

Este un utilitar folosit în detectarea abuzării bufferelor:

- a. nasm
- b. vsodium
- c. nvim
- d. IDA

[Clear my choice](#)

d

Folosim payloadul 64 * “A” + “\x23\x57\x04\x04” pentru a exploata o vulnerabilitate de tipul buffer overflow și pentru a suprascrie adresa de return a funcției cu valoarea 0x04045723. La ce adresă (relativ față de ebp) se găsește bufferul având în vedere că avem un sistem de 32 de biți?

Folosim payloadul $64 * "A" + "\x23\x57\x04\x04"$ pentru a exploata o vulnerabilitate de tipul buffer overflow și pentru a suprascrie adresa de return a funcției cu valoarea $0x04045723$. La ce adresă (relativ față de ebp) se găsește bufferul, având în vedere că avem un sistem pe 32 de biți?

- a. ebp-60
- b. ebp-68
- c. ebp-56
- d. ebp-64

1
Finish

În urma unui out of bounds(OOB) pe stivă:

Question 2
Not yet answered
Marked out of 1.00
 Flag question

Time left 0:07:53

În urma unui out of bounds (OOB) pe stivă:

a. Întotdeauna programul generează Segment Fault

b. Execuția programului nu este afectată

c. Se poate face disclose sau overwrite la adresa de return

d. Stiva se redimensionează

[Clear my choice](#)

[Next page](#)

Se dă funcția cu număr variabil de parametri "int magic(int n, ...)" și apelul "magic (3, 20, 14, 6)". Știind că adresa de return se află la adresa Y, la ce adresă se află parametrul 14 pe o arhitectură de x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

Se dă funcția cu număr variabil de parametri "int magic(int n, ...)" și apelul "magic(3, 20, 14, 6)". Știind că adresa de return se află la adresa Y, la ce adresă se află parametrul 14 pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

- a. Y + 12
- b. Y + 4
- c. Y + 8
- d. Y + 16

[Clear my choice](#)

Ce payload poate fi folosit pentru a rescrie adresa de return a unei funcții cu valoarea $0x01234567$ (exploatând o vulnerabilitate de tipul buffer overflow), știind că un buffer începe la adresa $ebp-40$?

Question 5
Not yet answered
Marked out of 1.00
 Flag question

Time left 0:01:41

Ce payload poate fi folosit pentru a rescrie adresa de return a unei funcții cu valoarea $0x01234567$ (exploatând o vulnerabilitate de tipul buffer overflow), știind că un buffer începe la adresa $ebp-40$?

a. 'Z'*48 + "\x67\x45\x23\x01"

b. 'Z'*40 + "\x01\x45\x23\x67"

c. 'Z'*44 + "\x67\x45\x23\x01"

d. 'Z'*40 + "\x01\x23\x45\x67"

[Clear my choice](#)

c) z*44

În cazul în care avem două fișiere: a.c și b.asm și dorim să obținem un executabil din cele două surse, ce utilitare vor fi folosite în acest proces?

În cazul în care avem două fișiere: a.c și b.asm și dorim să obținem un executabil din cele două surse, ce utilitare vor fi folosite în acest proces?

- a. Asamblor și compilator.
- b. Nu se poate obține executabil mixt.
- c. Compilator și linker.
- d. Compilator, asamblor și linker.

[Clear my choice](#)

Fie următorul cod: char buffer[100]; fgets(buffer, len, stdin); Ce proprietate trebuie să aibă len pentru a NU se realiza buffer overflow?

Fie următorul cod: char buffer[100]; fgets(buffer, len, stdin); Ce proprietate trebuie să aibă len pentru a NU se realiza buffer overflow?

- a. len > 100
- b. len = 100
- c. len <= 100
- d. Nu există o condiție pentru valoarea lui len

a) len > 100

Un buffer incepe la adresa ebo-92. Cat este offsetul necesar pentru a suprascrie adresa de return, pe un sistem pe 32 de biți?

Un buffer începe la adresa ebp-92. Cât este offsetul necesar pentru a suprascrie adresa de return, pe un sistem pe 32 de biți?

- a. 88
- b. 104
- c. 100
- d. 96

[Clear my choice](#)

96

Se da urmatoarea declaratie a unui buffer: char buf[64]. Care dintre urmatoarele variante NU poate genera un buffer overflow?

Se dă următoarea declarație a unui buffer: char buf[64]. Care dintre următoarele variante NU poate genera un buffer overflow?

- a. read(STDIN_FILENO, buf, 30)
- b. fgets(buf, 128, stdin)
- c. memcpy(buf, info, 256)
- d. write(STDOUT_FILENO, buf, 72)

[Clear my choice](#)

a

Ce date despre procedura curentă NU sunt puse pe stivă pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

Question 4
Not yet answered
Marked out of 0.00
Flag question

Ce date despre procedura curentă NU sunt puse pe stivă pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

- a. variabilele locale
- b. parametrii
- c. vechiul esp
- d. adresa de return

C. vechiul esp

Pe arhitectura x86 (32 de biți), în urma apelării unei funcții scrise în limbaj de asamblare, din cadrul unui program C, parametrii de pe stivă sunt eliberați de:

1
Not yet answered
Marked out of 0.00
Flag question

Pe arhitectura x86 (32 de biți), în urma apelării unei funcții scrise în limbaj de asamblare, din cadrul unui program C, parametrii de pe stivă sunt eliberați de:

- a. în mod implicit de către codul generat de compilator
- b. programatorul trebuie să specifice
- c. se eliberează toti la finalul programului
- d. funcția chemată

a) În mod implicit de către codul generat de compilator

Cu ajutorul payloadului `16 * "A" + "\x56\x34\x12\x07"` este suprascrisă adresa de return a unei funcții (exploatand o vulnerabilitate de tipul buffer overflow). La ce adresa relativă fata de registrul ebp se află bufferul? Am auzit ca **ebp-12**

Cu ajutorul payloadului `16 * "A" + "\x56\x34\x12\x07"` este suprascrisă adresa de return a unei funcții (exploatând o vulnerabilitate de tipul buffer overflow). La ce adresă relativă față de registrul `ebp` se află bufferul?

- a. `ebp-20`
- b. `ebp-8`
- c. `ebp-12`
- d. `ebp+4`

Care dintre următoarele poate fi considerat un DEZAVANTAJ al analizei dinamice?

Question 4

Not yet
answered

Marked out of
1.00

Flag
question

Care dintre următoarele poate fi considerat un DEZAVANTAJ al analizei dinamice?

- a. nu poate fi realizată pentru orice aplicație (inflexibilitate)
- b. nu se poate realiza analiză dacă n-avem acces la codul sursă
- c. nu poate detecta vulnerabilități apărute la runtime (de exemplu segmentation fault)
- d. în general, acoperirea (coverage) este mai redusă decât în cazul analizei statice

d) in general, acoperirea

Care va fi rezultatul în urma compilării, linkării și rularii următoarei secvențe de cod, în cazul în care nu mai există niciun alt modul?

Care va fi rezultatul în urma compilării, linkării și rularii următoarei secvențe de cod, în cazul în care nu mai există niciun alt modul?

```
#include <stdio.h>
extern int x;
int main(void) {
    x++;
    printf("%d", x);
    return 0;
}
```

- a. Programul va compila și va afișa o valoare aleatoare în funcție de ce se află la adresa variabilei `x`.
- b. Programul nu va compila deoarece variabila `x` este doar declarată, nu și definită.
- c. Programul va compila și va afișa valoarea 1.
- d. Programul nu va compila deoarece variabila `x` este doar definită, nu și declarată.

b?

Fie semnatura de funcție “`void funt(int a, int b, int c)`”. La ce offset față de `ebp` se află parametrii `a`, respectiv `c` pe un sistem pe 32 de biți?

Question 3

Not yet
answered

Marked out of
1.00

Flag
question

Fie semnătura de funcție “`void func(int a, int b, int c)`”. La ce offset față de `ebp` se află parametrii `a`, respectiv `c` pe un sistem pe 32 de biți?

- a. -8, -16
- b. +4, +12
- c. +8, +16
- d. -4, -12

+8, +16

Care dintre următoarele unelte este potrivita pentru a vizualiza codul binar și cel de asamblare al unui executabil?

Question 2
Not yet
answered
Marked out of
1.00
 Flag question

Care dintre următoarele unele este potrivită pentru a vizualiza codul binar și cel de asamblare al unui executabil?

- a. nm
- b. readelf
- c. objdump
- d. strings

[Clear my choice](#)

c

Care din următoarele este considerată o metoda de protecție împotriva atacurilor de tipul buffer overflow?

Question 5
Not yet
answered
Marked out of
1.00
 Flag
question

Care din următoarele este considerată o metodă de protecție împotriva atacurilor de tipul buffer overflow?

- a. Valori de tip canar (canary values)
- b. Address Space Layout Randomization
- c. Toate răspunsurile sunt corecte
- d. Bounds checking

Reprezinta o metoda de protectie împotriva overflow:

Reprezintă o metodă de protecție împotriva overflow:

- a. Information leak
- b. Canari
- c. Memory disclosure
- d. OOB (out of bounds)

B?

- probabil

https://en.wikipedia.org/wiki/Buffer_overflow_protection ...

⋮

Buffer overflow protection - Wikipedia

A **canary** example — Random XOR **canaries** have the same vulnerabilities as random **canaries**, except that the "read from stack" method of getting the **canary** is a ...

[Overview](#) · [Canaries](#) · [Tagging](#) · [Implementations](#)

Care dintre următoarele reprezinta o unealta de analiza dinamica?

Question 2



Not yet
answered

Marked out of
1.00

Flag
question

Care dintre următoarele reprezintă o unealtă de analiză dinamică?

- a. strings
- b. gcc
- c. gdb
- d. readelf

Gasit sub alta formă:

Care dintre următoarele utilitare este folosit pentru analiză dinamică? **GDB**

Un buffer incepe la adresa ebp-92. Cat este offsetul necesar pentru a suprascrie adresa de return, pe un sistem pe 32 de biți? Am auzit ca **96**

Question 4

Not yet
answered

Marked out of
1.00

Flag question

Un buffer incepe la adresa ebp-92. Cât este offsetul necesar pentru a suprascrie adresa de return, pe un sistem pe 32 de biți?



- a. 96
- b. 100
- c. 88
- d. 104

Ce date despre procedura curentă sunt puse pe stiva pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

Question 1

Not yet
answered

Marked out of
1.00

Flag question

Ce date despre procedura curentă sunt puse pe stivă pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

- a. variabilele globale
- b. parametrii
- c. vechiul esp
- d. ebp-ul funcției apelate

[Clear my choice](#)

Când poate apărea o eroare de tipul OOB (out of bounds)?

Question 4

Not yet
answered

Marked out of
1.00

Flag question

Când poate apărea o eroare de tipul OOB (out of bounds)?

- a. Când nu s-a apelat ret în cadrul unei funcții
- b. Când se fac operații dincolo de limita bufferului
- c. Când se face bounds checking
- d. Când programul nu afișează rezultatul așteptat

[Clear my choice](#)

b) Când se fac operații dincolo de limita bufferului

Limbajul C permite folosirea functiilor cu numar variabil de parametri.

Limbajul C permite folosirea funcțiilor cu număr variabil de parametri.

- a. Adevărat
- b. Fals

[Clear my choice](#)

a. adevarat

Este un utilitar folosit în detectarea abuzării bufferelor:

Question 2

Not yet
answered

Marked out of
1.00

[Flag question](#)

Este un utilitar folosit în detectarea abuzării bufferelor:

- a. nasm
- b. IDA
- c. nvim
- d. vsodium

IDA

Care dintre urmatoarele reprezinta un dezavantaj al folosirii limbajului

Question 1

Not yet
answered

Marked out of
1.00

[Flag question](#)

Care dintre următoarele reprezintă un dezavantaj al folosirii limbajului de asamblare, comparat cu folosirea unui limbaj de nivel mai înalt?

- a. acces la operații low-level
- b. mențenanță
- c. slab control asupra memoriei programului
- d. memoria ocupată de cod

[Clear my choice](#)

Se dau urmatoarele declaratii char buf[32], char long_string[256]. Care dintre urmatoarele apeluri de functii poate conduce la un buffer overflow?

Se dau următoarele declarații char buf[32], char long_string[256]. Care dintre următoarele apeluri de funcții poate conduce la un buffer overflow?

- a. fputs(long_string, sizeof(buf), stdout)
- b. memcpy(buf, long_string, sizeof(long_string))
- c. read(STDIN_FILENO, buf, 31)
- d. memcpy(buf, long_string, sizeof(buf))

memcpy(buf, long_string, sizeof(long_string))

Se da urmatoarea declaratie a unui buffer: char buf[64]. Care dintre urmatoarele variante NU poate genera un buffer overflow? Am auzit ca **read(STDIN_FILENO, buf, 30)**

Question 4

Not yet
answered

Marked out of
1.00

[Flag question](#)

Se dă următoarea declaratie a unui buffer: char buf[64]. Care dintre următoarele variante NU poate genera un buffer overflow?

- a. memcpy(buf, info, 256)
- b. read(STDIN_FILENO, buf, 30)
- c. fgets(buf, 128, stdin)
- d. write(STDOUT_FILENO, buf, 72)

În convenția de apel CDECL pe x86 (32 de biți), care dintre următoarele NU este o sarcină ce tine de apelant (caller)?

În convenția de apel CDECL pe x86 (32 de biți), care dintre următoarele NU este o sarcină ce ține de apelant (caller)?

- a. conservă EAX, ECX, EDX (dacă este cazul)
- b. aloca și eliberează variabilele locale
- c. plasează parametrii pe stivă
- d. curăță stiva după return

a) Conserva EAX

Alegeți varianta corectă:

Alegeți varianta corectă:

- a. Bufferele conțin informații temporare
- b. Tipul elementelor unui buffer este relevant
- c. Bufferele sunt simple variabile întregi
- d. Bufferele nu pot fi alocate dinamic

Bufferele conțin informații temporare

People also ask :

Is buffer a temporary memory?

In computer science, a data buffer (or just buffer) is a region of a physical memory storage used **to temporarily store data** while it is being moved from one place to another.

Conform convenției de apel CDECL, o valoare de return pe 64 de biți se pune:

Conform convenției de apel CDECL, o valoare de return pe 64 de biți se pune:

- a. în EDX:EAX
- b. în ESP
- c. nicio variantă nu este corectă
- d. în EDX:EBX

[Clear my choice](#)

a. În EDX:EAX

Reprezinta o metoda de protectie impotriva overflow:

Reprezintă o metodă de protecție împotriva overflow:

- a. Information leak
- b. Canari
- c. Memory disclosure
- d. OOB (out of bounds)

n

Care dintre urmatoarele este potrivita pentru a vizualiza codul binar si cel de asamblare al unui executabil?

Care dintre următoarele unelte este potrivită pentru a vizualiza codul binar și cel de asamblare al unui executabil?

- a. strings
- b. objdump
- c. readelf
- d. nm

[Clear my choice](#)

Care dintre urmatoarele este o secvență inline assembly

Care dintre următoarele este o secvență inline assembly ce produce o eroare la compilarea programului C?

- a. __asm__ ("mov ebx, %1\n\t"
"mov eax, %0\n\t"
"sub eax, ebx\n\t"
: "r" (val)
: "r" (no)
: "ebx"
);
- b. __asm__ ("mov %0, eax\n\t": "=r" (val));
- c. __asm__ ("add eax, ebx\n\t"
: "=a" (res)
: "a" (arg1), "b" (arg2)
);
- d. int no = 100, val ;
__asm__ ("mov ebx, %1\n\t"
"mov %0, ebx\n\t"
: "=r" (val)
: "r" (no) : "ebx"
);

b

Care etapa a compilarii permite obtinerea unui executabil mixt din doua sau mai multe fisiere obiect? (C-ASM)

Care etapă a compilării permite obținerea unui executabil mixt din două sau mai multe fișiere obiect? (C-ASM)

- a. asamblare
- b. acest lucru nu este posibil
- c. link-editare
- d. compilare

link-editare

Cu ce directivă trebuie declarat, în limbaj de asamblare, label-ul unei proceduri ce va fi ulterior chemată într-un program C?

Time left: 00:00:00

Cu ce directivă trebuie declarat, în limbaj de asamblare, label-ul unei proceduri ce va fi ulterior chemată într-un program C?

- a. nu este nevoie de label
- b. global
- c. asm
- d. extern

[Clear my choice](#)

Extern? E mai probabil

Care dintre următoarele variante de răspuns este adevarata cu privire la directiva de asamblare global?

Care dintre următoarele variante de răspuns este adevărată cu privire la directiva de asamblare global?

- a. Este obligatoriu să specificam tipul etichetei.
- b. Directiva global marchează o etichetă accesibilă și din alte module ale programului.
- c. Directiva global se folosește doar pentru variabile.
- d. Directiva global se folosește doar pentru nume de proceduri.

B, directiva global marcheaza...

Care afirmație este adevarata în cazul unui buffer overflow?

Care afirmație este adevărată în cazul unui buffer overflow?

- a. Se poate modifica fluxul normal de execuție al programului
- b. Se pot citi informații dincolo de limita bufferului
- c. Toate variantele sunt corecte
- d. Pot fi afectate informații dintr-un alt stack frame

C, toate variantele sunt corecte

Care dintre următoarele reprezintă un dezavantaj al folosirii limbajului de asamblare, comparat cu folosirea unui limbaj de nivel mai înalt?

Care dintre următoarele reprezintă un dezavantaj al folosirii limbajului de asamblare, comparat cu folosirea unui limbaj de nivel mai înalt?

- a. memoria ocupată de cod
- b. slab control asupra memoriei programului
- c. acces la operații low-level
- d. mențenanță

d) mententanta

În urma unui out of bounds (OOB) pe stiva:

În urma unui out of bounds (OOB) pe stivă:

- a. Întotdeauna programul generează Segmentation Fault
- b. Execuția programului nu este afectată
- c. Se poate face disclose sau overwrite la adresa de return
- d. Stiva se redimensionează

[Clear my choice](#)

C. probably

Alegeți varianta corecta:

Alegeți varianta corectă:

- a. Bufferele conțin informații temporare
- b. Tipul elementelor unui buffer este relevant
- c. Bufferele sunt simple variabile întregi
- d. Bufferele nu pot fi alocate dinamic

a) Bufferele conțin informații temporare

Se dă funcția cu număr variabil de parametri ... magic

Se dă funcția cu număr variabil de parametri "int magic(int n, ...)" și apelul "magic(3, 20, 14, 6)". Știind că adresa de return se află la adresa Y, la ce adresă se află parametrul 14 pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

- a. Y + 12
- b. Y + 4
- c. Y + 8
- d. Y + 16

[Clear my choice](#)

Potrivit convenției CDECL, pe arhitectura x86 (32 de biți), parametrii unei funcții se plasează:

Potrivit convenției CDECL, pe arhitectura x86 (32 de biți), parametrii unei funcții se plasează:

- a. în registre consacrate
- b. pe stivă, în ordinea inversă a transmiterii argumentelor
- c. pe stivă, în ordinea transmiterii argumentelor
- d. în registrele aflate libere la acel moment

Care dintre următoarele este un motiv pentru folosirea limbajului de asamblare?

- a. oferă acces la hardware și control
- b. este un limbaj mai ușor de înțeles
- c. niciun răspuns anterior nu este corect
- d. este un limbaj mai lent, dar mai ușor de scris

Se dă declarația: int *array[128]. Ce tip are expresia: *(&array)[0]

- a. int *
- b. int
- c. int **
- d. Niciuna din variante

Care NU este un avantaj al folosirii unui limbaj de nivel înalt în locul folosirii unuia de nivel jos?

- a. Lucrul apropiat de hardware
- b. Portabilitatea
- c. Ușurința citirii codului
- d. Dezvoltarea mai rapidă

Pe ce arhitectură poate fi rulat codul scris în C, fără a fi compilat în cod mașină?

- a. Niciuna din variante
- b. ARM
- c. MIPS
- d. x86



Clear my choice

Ce produce următoarea secvență de cod: int main0 { int* arr; arr[5] = 10; fprintf(stdout, "%d", arr[5]); }

- a. nu se afișează nimic
- b. eroare la compilare
- c. rezultat incert, posibil segmentation fault
- d. se afișează 10

Care este un avantaj al utilizării linkării statice în locul celei dinamice?

- a. codul bibliotecilor este partajat între procese la runtime
- b. nu mai este necesară etapa de loading
- c. executabilul obținut poate rula pe orice configurație software
- d. nu este necesară linkarea cu biblioteca standard C

Care dintre următoarele afirmații despre limbajul C și limbajul de asamblare este adevărată?

- a. limbajul de asamblare este system-dependent, iar limbajul C este system-independent
- b. ambele sunt pe același nivel ierarhic de abstractizare
- c. ambele sunt system-independent
- d. ambele sunt system-dependent

Care sunt componentele arhitecturii von Neumann clasice?

- a. două unități centrale de procesare (CPU/UCP), o unitate de memorie, o unitate de input și o unitate de output
- b. o unitate CPU/UCP, o unitate de memorie, o unitate de input și o unitate de output
- c. o unitate de control și o unitate de memorie
- d. o unitate CPU/UCP, două unități de memorie, o unitate de input și o unitate de output

În timpul compilării unui program, în etapa de linkare se generează:

- a. Fișier ce conține cod mașină fără referințe la biblioteci
- b. Fișier ce conține cod în limbaj de asamblare
- c. Fișier ce conține cod mașină, dar care are bibliotecile linkate probabil
- d. Fișier ce conține cod C/C++

Care dintre următoarele este un motiv pentru folosirea limbajului de asamblare?

- a. oferă acces la hardware și control
- b. este un limbaj mai ușor de înțeles
- c. niciun răspuns anterior nu este corect
- d. este un limbaj mai lent, dar mai ușor de scris

Se dă declarația: int key[256]. Ce output va avea expresia: sizeof(key)?

- a. 1024
- b. 1000
- c. 512
- d. 256

Pe ce magistrală este pus semnalul de citire scriere în procesul de citire scriere de date din în RAM?

- a. I/O bus
- b. control bus
- c. data bus
- d. address bus

In timpul compilării unui program, în etapa de linkare se generează:

- a. Fișier ce conține cod în limbaj de asamblare
- b. Fișier ce conține cod C/C++
- c. Fișier ce conține cod mașină, dar care are bibliotecile linkate
- d. Fișier ce conține cod mașină fără referințe la biblioteci

[Clear my choice](#)

Care este un avantaj al utilizării linkării dinamice în locul celei statice?

- a. executabilul obținut are o dimensiune mai mică
- b. datele programului inițial nu mai sunt plasate în executabil
- c. portabilitatea executabilului obținut
- d. zona de cod a programului inițial nu mai este plasată în executabil

[Clear my choice](#)

Ciclul de execuție al oricărui procesor este următorul:

- a. fetch-decode-datafetch-execute-writeback
- b. fetch-decode-datafetch-writeback-execute
- c. fetch-datafetch-decode-execute-writeback
- d. datafetch-decode-fetch-execute-writeback

[Clear my choice](#)

Care dintre variantele de mai jos NU este un exemplu de arhitectură de procesor?

- a. Linux
- b. RISC-V
- c. MIPS
- d. ARM

[Clear my choice](#)

Ce produce următoarea secvență de cod: int main() { int* arr; arr[5] = 10; fprintf(stdout, "%d", arr[5]); }

- a. rezultat incert, posibil segmentation fault
- b. eroare la compilare
- c. se afișează 10
- d. nu se afișează nimic

[Clear my choice](#)

Care NU este un avantaj al folosirii unui limbaj de nivel înalt în locul folosirii unuia de nivel jos?

- a. Portabilitatea
- b. Lucrul apropiat de hardware
- c. Dezvoltarea mai rapidă
- d. Ușurința citirii codului

[Clear my choice](#)

Care este un avantaj al utilizării linkării statice în locul celei dinamice?

- a. zona de cod a programului inițial nu mai este plasată în executabil
- b. portabilitatea executabilului obținut
- c. codul bibliotecilor este partajat între procese la runtime
- d. executabilul obținut are o dimensiune mai mică

[Clear my choice](#)

Care este un avantaj al utilizării linkării dinamice în locul celei statice?

- a. nu este necesară linkarea cu biblioteca standard C
- b. nu mai este necesară etapa de loading
- c. executabilul obținut are o dimensiune mai mare
- d. codul bibliotecilor este partajat între procese la runtime

Ce linie de cod generează, cu siguranță, eroare la compilare?

- a. `a = **b;`
- b. `*a = *b;`
- c. `&a = 5;`
- d. `*a = 5;`

Care dintre opțiunile de mai jos NU reprezintă o responsabilitate a linkerului?

- a. rezolvarea simbolurilor
- b. relocarea simbolurilor
- c. unificarea secțiunilor
- d. generarea de cod mașină

În timpul compilării unui program, etapa de preprocesare are rolul:

- a. Sa înlocuiască macro-urile
- b. Să șteargă comentariile
- c. Să înlocuiască #include<file> cu fișierul respectiv
- d. Toate variantele sunt corecte

Care dintre următoarele afirmații este adevărată?

- a. nu există nicio diferență între execuția pipeline și execuția secvențială
- b. execuția pipeline este mai rapidă decât cea secvențială deoarece aceasta nu folosește toate instrucțiunile dintr-un ciclu de execuție (nu face writeback)
- c. execuția pipeline este folosită pentru a realiza mai multe instrucțiuni într-un timp mai scurt, comparativ cu execuția secvențială **probabel**
- d. execuția secvențială este folosită pentru a realiza mai multe instrucțiuni decât reușește execuția pipeline

[Clear my choice](#)

Ce linie de cod generează, cu siguranță, eroare la compilare?

- a. `*a = 10;`
- b. `a = **b;`
- c. `a = &10;`
- d. `*a = *b;`

[Clear my choice](#)

Started on Friday, 9 October 2020, 6:56 PM

State Finished

Completed on Friday, 9 October 2020, 6:57 PM

Time taken 44 secs

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care sunt etapele compilării (în ordine) a unui program de la cod sursă la executabil?

- a. Preprocesare, asamblare, compilare, linkare
- b. Compilare, preprocesare, linkare, asamblare
- c. Preprocesare, compilare, asamblare, linkare
- d. Preprocesare, compilare, linkare, asamblare



The correct answer is: Preprocesare, compilare, asamblare, linkare

Question **2**

Correct

Mark 1.00 out of 1.00

Care dintre opțiunile de mai jos NU reprezintă un avantaj al folosirii unui compilator?

- a. Oferă portabilitate între diferite arhitecturi de procesor
- b. Generează automat cod sursă din descrierea în limbaj natural
- c. Optimizează codul scris în limbaj de nivel înalt în măsura în care se poate
- d. Ascunde complexitatea codului mașină



The correct answer is: Generează automat cod sursă din descrierea în limbaj natural

Question **3**

Correct

Mark 1.00 out of 1.00

Care NU este un avantaj al folosirii unui limbaj de nivel înalt în locul folosirii unuia de nivel jos?

- a. Portabilitatea
- b. Lucrul apropiat de hardware
- c. Ușurința citirii codului
- d. Dezvoltarea mai rapidă



The correct answer is: Lucrul apropiat de hardware

Question 4

Correct

Mark 1.00 out of 1.00

Care dintre variantele de mai jos NU este un exemplu de arhitectură de procesor?

- a. PowerPC
- b. macOS
- c. ARM
- d. MIPS



The correct answer is: macOS

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre opțiunile de mai jos NU reprezintă un pas pe care îl face procesorul în momentul rulării unui program?

- a. Tratarea erorilor de sintaxă apărute în programe
- b. Preluarea instrucțiunilor din memorie
- c. Decodificarea instrucțiunilor
- d. Executarea instrucțiunilor decodificate



The correct answer is: Tratarea erorilor de sintaxă apărute în programe

[◀ Tema 4](#)[Jump to...](#)[Curs 0 ►](#)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [12 October - 18 October](#) / [Lucrare de curs 2: ASC + Arhitectura x86](#)

Started on Friday, 16 October 2020, 4:15 PM

State Finished

Completed on Friday, 16 October 2020, 4:17 PM

Time taken 2 mins 3 secs

Marks 3.00/5.00

Grade **6.00** out of 10.00 (60%)

Question 1

Correct

Mark 1.00 out of 1.00

Care sunt pașii pentru crearea unui fișier executabil plecând de la un cod sursă?

- a. compilare, preprocesare, asamblare, link-editare
- b. preprocesare, compilare, link-editare, asamblare
- c. preprocesare, asamblare, link-editare, compilare
- d. preprocesare, compilare, asamblare, link-editare



The correct answer is: preprocesare, compilare, asamblare, link-editare

Question 2

Incorrect

Mark 0.00 out of 1.00

Care este relația dintre latența memoriilor și capacitatea lor?

- a. cu cât latența este mai mica, capacitatea este mai mica
- b. cu cât latența este mai mare, capacitatea este mai mare
- c. cu cât latența este mai mica, capacitatea este mai mare
- d. sunt independente



The correct answer is: cu cât latența este mai mica, capacitatea este mai mare

Question 3

Incorrect

Mark 0.00 out of 1.00

Prin intermediul cărei componente se realizează stocarea de date?

- a. procesor
- b. I/O
- c. CPU
- d. RAM



The correct answer is: RAM

Question 4

Correct

Mark 1.00 out of 1.00

Alegeți afirmația corectă în ceea ce privește limbajele de nivel înalt:

- a. programele nu sunt portabile
- b. dezvoltarea este mai lentă
- c. dezvoltarea este mai rapidă
- d. mențenanța este dificilă



The correct answer is: dezvoltarea este mai rapidă

Question 5

Correct

Mark 1.00 out of 1.00

Ce se întâmplă dacă se încearcă accesarea unei zone de memorie, printr-o adresă nevalidă (care nu a fost alocată în prealabil) ?

- a. programul funcționează corect
- b. eroare la compilare
- c. segmentation fault
- d. adresa respectivă devine validă



The correct answer is: segmentation fault

[◀ Curs 3](#)[Jump to...](#)[Laborator 1 ►](#)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [19 October - 25 October](#) / [Lucrare de curs 3: Arhitectura x86](#)

Started on Monday, 19 October 2020, 3:27 PM

State Finished

Completed on Monday, 19 October 2020, 3:32 PM

Time taken 4 mins 31 secs

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele zone din memoria unui program crește "în jos" (se folosește de la adrese mari la adrese mici)?

- a. Heap-ul
- b. Stiva ✓
- c. Segmentul de date
- d. Segmentul BSS

The correct answer is: Stiva

Question **2**

Correct

Mark 1.00 out of 1.00

Ce registru NU poate fi modificat direct (adică NU poate fi explicit modificat ca parte a unei instrucțiuni)?

- a. ESI
- b. DH
- c. EAX
- d. EIP ✓

The correct answer is: EIP

Question **3**

Correct

Mark 1.00 out of 1.00

După execuția unei instrucțiuni arbitrară, ce registru va fi întotdeauna afectat?

- a. ESP
- b. EIP ✓
- c. EDI
- d. EBP

The correct answer is: EIP

Question 4

Correct

Mark 1.00 out of 1.00

Care dintre următoarele registre este considerat un registru pointer (ține adesea o adresă)?

- a. EFLAGS
- b. ECX
- c. BH
- d. EBP



The correct answer is: EBP

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre următoarele registre mai este numit și registru acumulator?

- a. EDI
- b. ESI
- c. EBX
- d. EAX



The correct answer is: EAX

[◀ 2CC: Recap 3: Arhitectura sistemului de calcul](#)

Jump to...

[Lucrare de curs 4: Reprezentarea Datelor în Sistemele de Calcul ►](#)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [19 October - 25 October](#)

/ [Lucrare de curs 4: Reprezentarea Datelor în Sistemele de Calcul](#)

Started on Friday, 23 October 2020, 6:04 PM

State Finished

Completed on Friday, 23 October 2020, 6:07 PM

Time taken 3 mins 1 sec

Marks 4.00/5.00

Grade **8.00** out of 10.00 (**80%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care operație este echivalentă cu împărțirea numărului n cu 16?

- a. $n \gg 4$
- b. $n \gg 16$
- c. $n << 16$
- d. $n << 4$



The correct answer is: $n \gg 4$

Question **2**

Correct

Mark 1.00 out of 1.00

Fie variabila unsigned char n = 255. Ce valoare va avea ea dacă îi facem cast la signed char?

- a. eroare de compilare
- b. -1
- c. 255
- d. 0



The correct answer is: -1

Question 3

Correct

Mark 1.00 out of 1.00

Pe câți biți se reprezintă o variabilă de tip long long?

- a. 16 biți
- b. 64 de biți
- c. 32 de biți
- d. 8 biți



The correct answer is: 64 de biți

Question 4

Correct

Mark 1.00 out of 1.00

Cum putem verifica că a și b diferă printr-un singur bit?

- a. $a \wedge b > 0$
- b. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \& b$
- c. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$
- d. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \mid b$



The correct answer is: $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$

Question 5

Incorrect

Mark 0.00 out of 1.00

Care dintre următoarele condiții este necesară ca numerele nenule x și y să aibă semne diferite?

- a. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \& b$
- b. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$
- c. $a \wedge b > 0$
- d. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \mid b$



The correct answer is: $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$

[◀ Lucrare de curs 3: Arhitectura x86](#)

Jump to...

Laborator 3 ►

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [26 October - 1 November](#)

/ [Lucrare de curs 5: Reprezentarea Datelor în Sistemele de Calcul - C2](#)

Started on Friday, 30 October 2020, 6:53 PM

State Finished

Completed on Friday, 30 October 2020, 6:57 PM

Time taken 3 mins 32 secs

Marks 4.00/5.00

Grade **8.00** out of 10.00 (**80%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care este domeniul de valori ce poate fi stocat de o variabilă de tip short?

- a. [-2¹⁶, 2¹⁶]
- b. [0, 2¹⁶]
- c. [-16, 16]
- d. [-2¹⁵, 2¹⁵ - 1]



The correct answer is: [-2¹⁵, 2¹⁵ - 1]

Question **2**

Correct

Mark 1.00 out of 1.00

Cât înseamnă 0xFF în reprezentarea fară semn?

- a. -1
- b. 255
- c. -127
- d. -255



The correct answer is: 255

Question 3

Correct

Mark 1.00 out of 1.00

Pe câți biți se reprezintă un vector de tip unsigned char cu 3 elemente?

- a. 24 biți
- b. 48 de biți
- c. 8 biți
- d. 36 de biți



The correct answer is: 24 biți

Question 4

Correct

Mark 1.00 out of 1.00

Fie variabila char n = -100. Ce valoare va avea ea dacă îl facem cast la unsigned char?

- a. 227
- b. eroare de compilare
- c. 124
- d. 156



The correct answer is: 156

Question 5

Incorrect

Mark 0.00 out of 1.00

Fie declarația char s[] = "ANA"; Știind că A = 0x41, N=0x4E în ce se termina sirul declarat s?

- a. 0x1
- b. 0x41
- c. 0x4E
- d. 0x00



The correct answer is: 0x00

[◀ 2CC: Recap 5: Reprezentarea datelor în sistemul de calcul \(partea a 2-a\)](#)

Jump to...

Curs 7 ►

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [2 November - 8 November](#) / [Lucrare de curs 6: Setul de instructiuni](#)

Started on Monday, 2 November 2020, 12:42 PM

State Finished

Completed on Monday, 2 November 2020, 12:47 PM

Time taken 5 mins 1 sec

Marks 1.00/5.00

Grade **2.00** out of 10.00 (20%)

Question **1**

Incorrect

Mark 0.00 out of 1.00

În ce registru este stocat restul împărțirii pentru un împărțitor pe 16 biți (de tip word)?

- a. AH
- b. AX
- c. DX
- d. EBX

✗

The correct answer is: DX

Question **2**

Incorrect

Mark 0.00 out of 1.00

Ce instrucțiune nu afectează niciun indicator de stare?

- a. push
- b. cmp
- c. dec
- d. shl

✗

The correct answer is: push

Question **3**

Correct

Mark 1.00 out of 1.00

În urma operației "rol AL, 3", rezultatul din registrul AL este 0x5B. Cât era valoarea inițială din AL?

- a. 0x6B
- b. 0xDA
- c. 0xB6
- d. 0xAD

✓

The correct answer is: 0x6B

Question 4

Incorrect

Mark 0.00 out of 1.00

În urma operației de shiftare, cât va fi rezultatul din registrul AL și ce valoare va avea CF (Carry Flag)?

mov AL, 0x76

shr AL, 5

- a. Valoarea din AL este 0xC0, iar CF = 1.
- b. Valoarea din AL este 0xC0, iar CF = 0. ✗
- c. Valoarea din AL este 0x03, iar CF = 0.
- d. Valoarea din AL este 0x03, iar CF = 1.

The correct answer is: Valoarea din AL este 0x03, iar CF = 1.

Question 5

Incorrect

Mark 0.00 out of 1.00

Ce flag se activează în urma instrucțiunilor de mai jos?

mov AL, 100

add AL, 45

- a. Zero Flag
- b. Carry Flag
- c. Overflow Flag
- d. Parity Flag ✗

The correct answer is: Overflow Flag

◀ Laborator 4

Jump to...

Lucrare de curs 7: Declararea datelor ►

Started on Friday, 6 November 2020, 8:32 PM

State Finished

Completed on Friday, 6 November 2020, 8:36 PM

Time taken 3 mins 16 secs

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele instrucțiuni poate fi folosită pentru a implementa eficient funcția **strcpy** din C:

- a. cmpsb
- b. scasb
- c. movsb
- d. lodsb



The correct answer is: movsb

Question **2**

Correct

Mark 1.00 out of 1.00

Adevărat sau fals: Folosirea instrucțiunilor pentru șiruri (**movs**, **scas** etc.) este **ÎNTOTDEAUNA** (pe orice procesor cu arhitectura x86, cuplat la orice periferice) mai rapidă decât folosirea instrucțiunilor generale (**mov**, **cmp** etc.).

- a. Fals
- b. Adevărat



The correct answer is: Fals

Question **3**

Correct

Mark 1.00 out of 1.00

Care este diferența dintre instrucțiunile **sar** și **shr** în limbajul de asamblare x86?

- a. **sar** păstrează valoarea bitului celui mai **SEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 1
- b. **sar** păstrează valoarea bitului celui mai **NESEMNICATIV** al operandului destinație, pe când **shr** îl transformă în 0
- c. **sar** păstrează valoarea bitului celui mai **SEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 0
- d. **sar** păstrează valoarea bitului celui mai **NESEMNICATIV** al operandului destinație, pe când **shr** îl transformă în 1



The correct answer is: **sar** păstrează valoarea bitului celui mai **SEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 0

Question 4

Correct

Mark 1.00 out of 1.00

Cum putem modifica direcția în care sunt parcurse de către instrucțiunea **movsb** șirurile ale căror adrese sunt stocate în **ESI** și **EDI**?

- a. Direcția de parcurgere a șirurilor este mereu "de la stânga la dreapta", adică în sensul creșterii adreselor din **ESI** și **EDI**
- b. Folosind instrucțiunile **std** sau **cld**
- c. Prefixand instrucțiunea **movsb** cu una dintre instrucțiunile **repe** sau **repne**
- d. Folosind instrucțiunea **cdq**



The correct answer is: Folosind instrucțiunile **std** sau **cld**

Question 5

Correct

Mark 1.00 out of 1.00

O linie de cod în limbaj de asamblare poate conține:

- a. Unul sau mai multe labeluri
- b. Cel mult o instrucțiune de jump
- c. Cel puțin o mnemonică
- d. Cel mult un operand



The correct answer is: Cel mult o instrucțiune de jump

[← Lucrare de curs 6: Setul de instrucțiuni](#)

Jump to...

[Laborator 5 ►](#)

Started on Sunday, 15 November 2020, 9:23 PM

State Finished

Completed on Sunday, 15 November 2020, 9:24 PM

Time taken 1 min

Marks 4.00/5.00

Grade **8.00** out of 10.00 (80%)

Question 1

Correct

Mark 1.00 out of 1.00

Care din următoarele reprezinta un exemplu de adresare BAZATĂ ȘI INDEXATĂ CU SCALĂ (bază + index * scala + deplasament)?

- a. mov eax, [edx + ebx*4 + 0x12345678]
- b. mov eax, [ebp + 8]
- c. mov eax, [0xabcddef00]
- d. mov eax, [esi + ebx + 0x8]



The correct answer is: mov eax, [edx + ebx*4 + 0x12345678]

Question 2

Correct

Mark 1.00 out of 1.00

Cum se declară corect o variabilă 'myVar' de tip BYTE neinițializată?

- a. myVar dw ?
- b. myVar db ?
- c. myVar db null
- d. myVar db undefined



The correct answer is: myVar db ?

Question 3

Correct

Mark 1.00 out of 1.00

Care din următoarele registre NU este registru general?

- a. SS
- b. ESI
- c. BH
- d. EBP



The correct answer is: SS

Question 4

Correct

Mark 1.00 out of 1.00

Alegeți specificatorul de tip invalid:

- a. HWORD (6 octeți)
- b. QWORD (8 octeți)
- c. DWORD (4 octeți)
- d. BYTE (1 octet)



The correct answer is: HWORD (6 octeți)

Question 5

Incorrect

Mark 0.00 out of 1.00

Care este instrucțiunea dintr-un macro care înmulțește cu 4 al doilea parametru al macro-ului?

- a. sal %2, 2
- b. sal %2, 4
- c. sal %1, 2
- d. mul %1, 4



The correct answer is: sal %2, 2

[◀ 2CC: Verificare: Moduri de adresare](#)

Jump to...

[Lucrare de curs 9: Moduri de adresare C2 + Stiva ►](#)

Started on Tuesday, 17 November 2020, 10:31 AM**State** Finished**Completed on** Tuesday, 17 November 2020, 10:33 AM**Time taken** 1 min 52 secs**Marks** 4.00/5.00**Grade** **8.00** out of 10.00 (80%)**Question 1**

Correct

Mark 1.00 out of 1.00

Cum se face transmiterea parametrilor unei funcții, în programele care folosesc sintaxa de apel cdecl pe o arhitectură x86 (32 de biți)?

- a. prin heap
- b. printr-un vector
- c. toate variantele sunt greșite
- d. prin stivă



The correct answer is: prin stivă

Question 2

Correct

Mark 1.00 out of 1.00

Să presupunem că avem o stivă goală și rulăm următoarea secvență de instrucțiuni: push 1; push 2; pop; push 3; pop; push 4; push 5; pop. Care va fi elementul din vârful stivei?

- a. 3
- b. 4
- c. 1
- d. 5



The correct answer is: 4

Question 3

Correct

Mark 1.00 out of 1.00

Care operație NU poate fi efectuată asupra stivei?

- a. pop
- b. index
- c. pushad
- d. push



The correct answer is: index

Question 4

Incorrect

Mark 0.00 out of 1.00

În general, care dintre variantele de mai jos este un dezavantaj al folosirii macrourilor?

- a. duc la o programare mai neficientă
- b. duc la o execuție mai lentă a programului
- c. se pot crea instrucțiuni noi
- d. fac codul mai greu de înțeles



The correct answer is: fac codul mai greu de înțeles

Question 5

Correct

Mark 1.00 out of 1.00

Cum se pot interschimba variabilele a și b folosind stiva?

- a. push b; push a; pop a; pop b
- b. push a; pop b; push b; pop a
- c. mov eax, a; mov a, b; mov b, eax
- d. push a; push b; pop a; pop b



The correct answer is: push a; push b; pop a; pop b

[◀ Lucrare de curs 8: Moduri de adresare](#)

Jump to...

[Curs 10-11 ►](#)

Started on Sunday, 22 November 2020, 11:57 AM

State Finished

Completed on Sunday, 22 November 2020, 12:01 PM

Time taken 4 mins 22 secs

Marks 1.00/5.00

Grade **2.00** out of 10.00 (20%)

Question **1**

Incorrect

Mark 0.00 out of 1.00

Ce va afișa funcția următoare, știind că a primit parametrii 1 și 2 în ordinea aceasta (push 1, push 2)?

myFunc:

```
pop eax
PRINTF32 `%d\x0`, eax
ret
```

- a. 2 ✗
- b. altă valoare
- c. nimic, pentru că va da eroare înainte de la instrucțiunea "pop"
- d. 1

The correct answer is: altă valoare

Question **2**

Incorrect

Mark 0.00 out of 1.00

Prin convenție, pe o arhitectură x86 (32 de biți), valoarea de return a unei funcții se plasează de regulă:

- a. în registrul esp
- b. în registrul eax (sau edx:eax, în cazul unui rezultat pe 8 octeți)
- c. pe stivă, urmând să fie preluată tot de acolo ✗
- d. într-o altă zonă de memorie (rodata, data, etc.)

The correct answer is: în registrul eax (sau edx:eax, în cazul unui rezultat pe 8 octeți)

Question 3

Incorrect

Mark 0.00 out of 1.00

Considerăm echivalentul în limbaj de asamblare al semnăturii următoare de funcție: "void myFunc(int arr[], int dim)". Dacă valoarea lui ebp este salvată pe stivă la începutul funcției, ce se găsește la adresa esp + 8?

- a. adresa de început a lui arr
- b. valoarea lui dim
- c. valoarea primului element din arr
- d. valoarea ultimului element din arr



The correct answer is: adresa de început a lui arr

Question 4

Incorrect

Mark 0.00 out of 1.00

Fie codul următor:

```
myFunc:
    push ebp
    mov ebp, esp
    sub esp, 2
    ...
```

La ce adresă se găsește valoarea primului parametru de pe stivă (pe o arhitectură x86)?

- a. esp + 10
- b. esp + 4
- c. esp + 6
- d. esp + 8



The correct answer is: esp + 10

Question 5

Correct

Mark 1.00 out of 1.00

Vrem să transmitem valoarea din registrul eax ca parametru unei funcții, fără a folosi instrucțiunea "push". O modalitate este:

- a. mov [esp], eax
 sub esp, 4
- b. sub esp, 4
 mov esp, eax
- c. mov esp, eax
 sub esp, 4
- d. sub esp, 4
 mov [esp], eax



The correct answer is: sub esp, 4
mov [esp], eax

[◀ Laborator 6](#)

Jump to...

[Laborator 7 ►](#)

Started on Friday, 27 November 2020, 5:28 PM

State Finished

Completed on Friday, 27 November 2020, 5:30 PM

Time taken 2 mins 8 secs

Marks 3.00/5.00

Grade **6.00** out of 10.00 (60%)

Question 1

Correct

Mark 1.00 out of 1.00

Fie următoarea semnătura de funcție "void func(struct point *p)" și structura:

```
struct point {
    int x;
    int y;
}
```

Cum se poate pune valoarea membrului y în registrul eax în interiorul funcției, presupunând că elementele structurii nu sunt modificate de compilator în privința ordinii și a spațiului dintre ele (padding)?

- a. mov eax, [ebp + 12]
 mov eax, [eax]
- b. mov eax, [ebp + 4]
 mov eax, [eax + 4]
- c. mov eax, [ebp + 8]
 mov eax, [eax + 4] ✓
- d. mov eax, [ebp + 8]
 mov eax, [eax + 1]

The correct answer is: mov eax, [ebp + 8]
mov eax, [eax + 4]

Question 2

Correct

Mark 1.00 out of 1.00

Ce date despre procedura curentă NU sunt puse pe stivă pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

- a. variabilele locale
- b. vechiul esp ✓
- c. parametrii
- d. adresa de return

The correct answer is: vechiul esp

Question 3

Incorrect

Mark 0.00 out of 1.00

Cu ce este echivalentă următoarea secvență în limbaj de asamblare?

```
push ebp
mov ebp, esp
sub esp, 4
```

- a. leave 4
- b. leave ✗
- c. enter 4, 0
- d. enter 1, 0

The correct answer is: enter 4, 0

Question 4

Incorrect

Mark 0.00 out of 1.00

Fie semnătura de funcție "void func(int a, int b, int c)". La ce offset față de ebp se află parametrii a, respectiv c pe un sistem pe 32 de biți?

- a. -4, -12 ✗
- b. +8, +16
- c. -8, -16
- d. +4, +12

The correct answer is: +8, +16

Question 5

Correct

Mark 1.00 out of 1.00

Fie următoarea funcție:

```
int func(int *arr, int size) {
    short *n;
    char *str;
    int y;
    ...
}
```

Știind că variabila n se află la adresa X, la ce adresă se află variabila y pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl? Considerăm că variabilele locale nu sunt reordonate de către compilator și că nu există niciun padding.

- a. X + 8
- b. X - 8 ✓
- c. X + 5
- d. X - 5

The correct answer is: X - 8

[◀ 2CC: Hammer Time](#)[Jump to...](#)[Lucrare de curs 12: C + asm ►](#)

Started on Wednesday, 2 December 2020, 5:57 PM**State** Finished**Completed on** Wednesday, 2 December 2020, 5:58 PM**Time taken** 50 secs**Marks** 4.00/5.00**Grade** **8.00** out of 10.00 (80%)**Question 1**

Correct

Mark 1.00 out of 1.00

Conform convenției de apel CDECL, o valoare de return pe 32 de biți se pune:

- a. în registrul ESP
- b. pe stivă
- c. în registrul EIP
- d. în registrul EAX



The correct answer is: în registrul EAX

Question 2

Correct

Mark 1.00 out of 1.00

Care dintre următoarele reprezintă un dezavantaj al folosirii limbajului de asamblare, comparat cu folosirea unui limbaj de nivel mai înalt?

- a. acces la operații low-level
- b. mentenanța
- c. memoria ocupată de cod
- d. slab control asupra memoriei programului



The correct answer is: mentenanța

Question 3

Correct

Mark 1.00 out of 1.00

Marcați cu adevărat (A) sau fals (B) afirmațiile: În general, inline assembly aduce optimizări programului scris în limbaj high-level/Numărul variabilelor folosibile în inline assembly este unul limitat

- a. A/F
- b. F/A
- c. F/F
- d. A/A



The correct answer is: A/A

Question 4

Incorrect

Mark 0.00 out of 1.00

Cu ce directivă trebuie prefixată declararea unui funcții simple C (antetul funcției), dar definită într-un alt modul C sau într-un modul în limbaj de asamblare?

- a. nu este nevoie de o directivă
- b. include
- c. global
- d. extern



The correct answer is: nu este nevoie de o directivă

Question 5

Correct

Mark 1.00 out of 1.00

Care registru NU poate fi modificat în mod direct de către programator?

- a. EIP
- b. EDI
- c. EBP
- d. ESP



The correct answer is: EIP

[◀ Lucrare de curs 11: Funcții \(2\)](#)

Jump to...

[Curs 13 ►](#)

Started on Friday, 11 December 2020, 9:54 PM

State Finished

Completed on Friday, 11 December 2020, 9:55 PM

Time taken 1 min

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele reprezintă o unealtă de analiză statică?

- a. objdump
- b. valgrind
- c. ld
- d. ltrace



The correct answer is: objdump

Question **2**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele variante de răspuns este adevărată cu privire la directiva de asamblare global?

- a. Directiva global marchează o etichetă accesibilă și din alte module ale programului.
- b. Este obligatoriu să specificam tipul etichetei.
- c. Directiva global se folosește doar pentru variabile.
- d. Directiva global se folosește doar pentru nume de proceduri.



The correct answer is: Directiva global marchează o etichetă accesibilă și din alte module ale programului.

Question 3

Correct

Mark 1.00 out of 1.00

Care dintre următoarele variante de răspuns este adevărată cu privire la unealta xxd?

- a. xxd poate fi folosit doar pentru codificarea în hexazecimal.
- b. xxd poate fi folosit atât pentru codificarea cât și pentru decodificarea în / din hexazecimal. ✓
- c. xxd poate fi folosit doar pentru decodificarea în hexazecimal.
- d. xxd poate fi folosit pentru eliminarea simbolurilor dintr-un executabil.

The correct answer is: xxd poate fi folosit atât pentru codificarea cât și pentru decodificarea în / din hexazecimal.

Question 4

Correct

Mark 1.00 out of 1.00

În care dintre următoarele situații unealta gdb se pretează cel mai bine?

- a. Vrem să aflăm simbolurile dintr-un executabil și ce siruri de caractere sunt definite în cadrul programului.
- b. Vrem să aflăm unde apare un memory leak.
- c. Vrem să aflăm valoarea registrelor într-un anume punct din execuția programului. ✓
- d. Vrem să aflăm mai multe metadate despre program.

The correct answer is: Vrem să aflăm valoarea registrelor într-un anume punct din execuția programului.

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre următoarele instrucțiuni poate fi folosită pentru a șterge simbolurile de debug dintr-un fișier binar?

- a. nm
- b. strings
- c. bc
- d. strip ✓

The correct answer is: strip

[◀ Laborator 8](#)

Jump to...

[2CC: Recap 14: Gestiunea bufferelor ►](#)

[Dashboard](#) / My courses / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [14 December - 20 December](#) / [Lucrare de curs 14: Buffer Overflow, Securitate](#)

Started on Sunday, 20 December 2020, 11:37 AM

State Finished

Completed on Sunday, 20 December 2020, 11:38 AM

Time taken 1 min 13 secs

Grade 8.00 out of 10.00 (80%)

Question 1

Correct

Mark 2.00 out of 2.00

NU reprezintă o metodă de abuzare a bufferelor:

- a. Memory disclosure
- b. OOB (out of bounds)
- c. Canari
- d. Buffer overflow



The correct answer is: Canari

Question 2

Correct

Mark 2.00 out of 2.00

Este un utilitar folosit în detectarea abuzării bufferelor:

- a. nasm
- b. IDA
- c. nvim
- d. vsodium



The correct answer is: IDA

Question 3

Correct

Mark 2.00 out of 2.00

Se dă următoarea declarație a unui buffer în limbaj de asamblare: sub esp, 256. Sistemul este pe 32 biți. Care dintre următoarele declarații C sunt echivalente declarației în limbaj de asamblare de mai sus?

- a. unsigned short[128], int buf[256], long long buf[8]
- b. double buf[32], int buf[64], unsigned short[128]
- c. int buf[32], char buf[256], double buf[16]
- d. int buf[64], unsigned short buf[128], char buf[128]



The correct answer is: double buf[32], int buf[64], unsigned short[128]

Question 4

Correct

Mark 2.00 out of 2.00

Când poate apărea o eroare de tipul OOB (out of bounds)?

- a. Când se face bounds checking
- b. Când nu s-a apelat ret în cadrul unei funcții
- c. Când programul nu afișează rezultatul așteptat
- d. Când se fac operații dincolo de limita bufferului



The correct answer is: Când se fac operații dincolo de limita bufferului

Question 5

Incorrect

Mark 0.00 out of 2.00

Fie următoarea funcție:`void func(void) { double trouble; int buf[10]; ... }` Câți octeți sunt între adresa de început a lui buf și adresa de return a lui f?

- a. Nu știm, depinde de compilator.
- b. 48
- c. 44
- d. 18



The correct answer is: Nu știm, depinde de compilator.

[◀ Laborator 9](#)

Jump to...

[Curs 16-17 ►](#)

Started on Monday, 11 January 2021, 6:31 PM

State Finished

Completed on Monday, 11 January 2021, 6:36 PM

Time taken 5 mins 1 sec

Marks 4.00/5.00

Grade **8.00** out of 10.00 (80%)

Question 1

Correct

Mark 1.00 out of 1.00

Un buffer începe la adresa ebp-92. Cât este offsetul necesar pentru a suprascrie adresa de return, pe un sistem pe 32 de biți?

- a. 88
- b. 100
- c. 104
- d. 96



The correct answer is: 96

Question 2

Incorrect

Mark 0.00 out of 1.00

În următorul payload: offset * "Z" + "\x78\x56\x34\x12" + 4 * "Y" + "\xAB\xCD\xEF\xGH" + "\xBE\xBA\xFE\xCA", buffer-ul găsindu-se la adresa ebp-offset. Ultima valoare reprezintă:

- a. valoarea primului parametru al funcției
- b. valoarea celui de-al doilea parametru al funcției
- c. adresa funcției
- d. adresa de return a funcției



The correct answer is: valoarea celui de-al doilea parametru al funcției

Question 3

Correct

Mark 1.00 out of 1.00

Care dintre următoarele utilitare este folosit pentru analiză dinamică?

- a. GDB
- b. IDA
- c. objdump
- d. nm



The correct answer is: GDB

Question 4

Correct

Mark 1.00 out of 1.00

Fie următorul cod: char buffer[100]; fgets(buffer, len, stdin); Ce proprietate trebuie să aibă len pentru a NU se realiza buffer overflow?

- a. len > 100
- b. len ≤ 100
- c. len = 100
- d. Nu există o condiție pentru valoarea lui len



The correct answer is: len ≤ 100

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre următoarele poate fi considerat un DEZAVANTAJ al analizei dinamice?

- a. nu poate fi realizată pentru orice aplicație (inflexibilitate)
- b. nu poate detecta vulnerabilități apărute la runtime (de exemplu segmentation fault)
- c. nu se poate realiza analiză dacă n-am acces la codul sursă
- d. în general, acoperirea (coverage) este mai redusă decât în cazul analizei statice



The correct answer is: în general, acoperirea (coverage) este mai redusă decât în cazul analizei statice

[◀ Laborator 10](#)

Jump to...

[Lucrare de curs 16: Optimizări ►](#)

Started on Saturday, 16 January 2021, 9:37 AM**State** Finished**Completed on** Saturday, 16 January 2021, 9:39 AM**Time taken** 1 min 58 secs**Marks** 5.00/5.00**Grade** **10.00** out of 10.00 (**100%**)**Question 1**

Correct

Mark 1.00 out of 1.00

Care afirmație este FALSĂ cu privire la instrucțiunile de tip jmp/branch?

- a. predicțiile făcute de CPU în cazul salturilor de tip branch sunt întotdeauna adevărate
- b. dacă avem doar o instrucțiune per loop, execuția datelor în pipeline nu mai are sens
- c. obținem îmbunătățiri dacă reducem din numărul instrucțiunilor de tip branch
- d. salturile la distanțe mari în cod fac imposibilă stocarea lui în cache



The correct answer is: predicțiile făcute de CPU în cazul salturilor de tip branch sunt întotdeauna adevărate

Question 2

Correct

Mark 1.00 out of 1.00

Care dintre următoarele transformări NU reprezintă o optimizare?

- a. pentru împărțirea mai multor numere la x, calculez $1/x$ și după multiplic
- b. folosirea insert sort, în locul lui qsort din libc
- c. folosirea instrucțiunilor de tipul CMOV în locul instrucțiunilor de tip branch
- d. folosirea shiftarilor în locul înmulțirii/împărțirii cu puteri ale lui 2



The correct answer is: folosirea insert sort, în locul lui qsort din libc

Question 3

Correct

Mark 1.00 out of 1.00

Care exemplu NU ilustrează aplicarea tehnicii loop unrolling?

- a. for ($i = 0; i < 100; i++$)

```
for (j = 0; j < 100; j++)  
    a[i][j] = b[i][j];
```

- b. for ($i = 0; i < 100; i++$)

```
for (j = 0; j < 100; j+=2) {  
    a[i][j] = b[i][j];  
    a[i][j+1] = b[i][j+1];  
}
```

- c. for ($i = 0; i < 100; i+=2$)

```
for (j = 0; j < 100; j++) {  
    a[i][j] = b[i][j];  
    a[i+1][j] = b[i+1][j];  
}
```

- d. for ($i = 0; i < 100; i+=2$)

```
for (j = 0; j < 100; j+=2) {  
    a[i][j] = b[i][j];  
    a[i][j+1] = b[i][j+1];  
    a[i+1][j] = b[i+1][j];  
    a[i+1][j+1] = b[i+1][j+1];  
}
```



The correct answer is: for ($i = 0; i < 100; i++$)

for ($j = 0; j < 100; j++$)

$a[i][j] = b[i][j];$

Question 4

Correct

Mark 1.00 out of 1.00

Cum pot fi optimizate operațiile matematice:

1. $x / 16$

2. x^4

3. x^3

- a. $x << 4 \quad | \quad x2 = \text{pow}(x, 2), x4 = \text{pow}(x2, 2) \quad | \quad x*x*x$

- b. $x >> 4 \quad | \quad \text{pow}(x, 4) \quad | \quad \text{pow}(x, 3)$

- c. $x >> 4 \quad | \quad x2 = x*x, x4 = x2*x2 \quad | \quad x*x*x$

- d. $x / \text{pow}(2, 4) \quad | \quad \text{pow}(x, 4) \quad | \quad x*x*x$



The correct answer is: $x >> 4 \quad | \quad x2 = x*x, x4 = x2*x2 \quad | \quad x*x*x$

Question 5

Correct

Mark 1.00 out of 1.00

Care afirmație este FALSĂ privind înmulțirea matricelor:

- a. deoarece capacitatea cache-ului este mare, este eficient dacă citim/încarcăm întreaga linie/coloană din matrice ✓
- b. algoritmul standard folosește 3 cicluri for imbricate, complexitatea fiind $O(n^3)$
- c. poate fi optimizat, folosind blocuri mai mici
- d. soluția optimizată folosește eficient memoria cache, mărindu-se altfel performanța programului

The correct answer is: deoarece capacitatea cache-ului este mare, este eficient dacă citim/încarcăm întreaga linie/coloană din matrice

[◀ Lucrare de curs 15: Buffer Overflow. Securitate \(2\)](#)

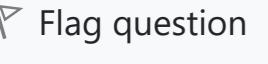
Jump to...

[Curs 18-19 ►](#)

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 11 January - 17 January / 2CC: Recap 15: Gestiunea bufferelor

Started on	Thursday, 14 January 2021, 4:05 PM
State	Finished
Completed on	Thursday, 14 January 2021, 4:07 PM
Time taken	1 min 49 secs
Marks	2.00/4.00
Grade	5.00 out of 10.00 (50%)

Question 1
Correct
Mark 1.00 out of 1.00


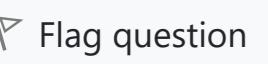
Pe un sistem pe 64 de biți, un buffer este la adresa rbp-72. Care este offsetul până la adresa de return?

- a. 88
- b. 76
- c. 72
- d. 80
- e. 64



Răspunsul dumneavoastră este corect.

The correct answer is: 80

Question 2
Correct
Mark 1.00 out of 1.00


Pe un sistem pe 32 de biți folosim payloadul $64*\'A' + p32(hidden_address)$ ca să suprascriem adresa de return cu adresa funcției `hidden_address`. La ce adresă este bufferul?

- a. `ebp+64`
- b. `ebp-8`
- c. `ebp+4`
- d. `ebp-60`
- e. `ebp-64`



Răspunsul dumneavoastră este corect.

The correct answer is: `ebp-60`

Question 3
Incorrect
Mark 0.00 out of 1.00

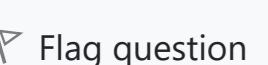

Un buffer este la adresa `ebp-40`, pe un sistem pe 32 de biți. Folosim payloadul $28*\'A' + p32(hidden_address)$. Ce vom suprascrie cu adresa `hidden_address`?

- a. un parametru al funcției
- b. fostul esp
- c. adresa de return
- d. fostul ebp
- e. o variabilă locală



Răspunsul dumneavoastră este incorrect.

The correct answer is: o variabilă locală

Question 4
Incorrect
Mark 0.00 out of 1.00


Un buffer este la adresa `ebp-32` pe un sistem pe 32 de biți. Dacă folosim funcția `fwrite()` pentru a scrie la standard output 56 de octeți de la adresa bufferului, ce se va întâmpla?

- a. vom face leak doar la adresa fostului `ebp`
- b. vom suprascrie doar fostul `ebp`
- c. vom face leak doar la adresa de return
- d. vom suprascrie fostul `ebp` și adresa de return
- e. vom face leak la adresa fostului `ebp` și adresa de return
- f. vom suprascrie adresa de return



Răspunsul dumneavoastră este incorrect.

The correct answer is: vom face leak la adresa fostului `ebp` și adresa de return

[Finish review](#)

◀ Curs 16-17

Jump to...



2CC: Recap 16: Optimizări ►

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 23 November - 29 November / 2CC: Recap 10: Funcții

Started on	Thursday, 26 November 2020, 4:06 PM
State	Finished
Completed on	Thursday, 26 November 2020, 4:08 PM
Time taken	1 min 28 secs
Marks	1.00/4.00
Grade	2.50 out of 10.00 (25%)

Question 1
Correct
Mark 1.00 out of 1.00
 Flag question

Ce NU stocăm pe stivă?

- a. variabilele locale
- b. variabilele globale
- c. parametrii funcției
- d. adresa de return
- e. fostul frame pointer



Răspunsul dumneavoastră este corect.

The correct answer is: variabilele globale

Question 2
Incorrect
Mark 0.00 out of 1.00
 Flag question

Cu ce este echivalentă instrucțiunea ret?

- a. mov eip, esp
- b. pop eip
- c. jmp eip
- d. push eip
- e. mov esp, eip



Răspunsul dumneavoastră este incorrect.

The correct answer is: pop eip

Question 3
Incorrect
Mark 0.00 out of 1.00
 Flag question

Ce registru este frame pointer pe arhitectura x86?

- a. eflags
- b. ebp
- c. esi
- d. esp
- e. eip



Răspunsul dumneavoastră este incorrect.

The correct answer is: ebp

Question 4
Incorrect
Mark 0.00 out of 1.00
 Flag question

Ce se găsește în vârful stivei la începutul execuției unei funcții?

- a. fostul frame pointer
- b. ultimul argument al funcției
- c. adresa următoarei instrucțiuni executate
- d. adresa de return
- e. primul argument al funcției



Răspunsul dumneavoastră este incorrect.

The correct answer is: adresa de return

[Finish review](#)

[◀ Laborator 7](#)

Jump to...



[2CC: Recap 11: Funcții ►](#)

You are logged in as Andrei STAN ([Log out](#))

[03-ACS-L-A2-S1-IOCLA-CTI](#)

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

Quiz navigation

- [1](#)
- [2](#)
- [3](#)
- [4](#)

[Finish review](#)



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 16 November - 22 November / 2CC: Verificare: Stiva. Funcții

Started on	Thursday, 19 November 2020, 5:04 PM
State	Finished
Completed on	Thursday, 19 November 2020, 5:06 PM
Time taken	2 mins 25 secs
Marks	2.00/4.00
Grade	5.00 out of 10.00 (50%)

Question 1
Incorrect
Mark 0.00 out of 1.00
 Flag question

Ce indică stack pointer-ul?

- a. adresa curentă în program
- b. instrucțiunea următoare
- c. adresa funcției care urmează să fie apelată
- d. istoricul de apeluri de funcții
- e. valoarea cumulată a registrelor



Răspunsul dumneavoastră este incorrect.

The correct answer is: istoricul de apeluri de funcții

Question 2
Incorrect
Mark 0.00 out of 1.00
 Flag question

Cum se numește fiecare element constituent din stiva unui program? Cum se numește fiecare element al stivei unui program?

- a. stack register
- b. stack frame
- c. back trace
- d. stack pointer
- e. stack heap



Răspunsul dumneavoastră este incorrect.

The correct answer is: stack frame

Question 3
Correct
Mark 1.00 out of 1.00
 Flag question

Cu ce este echivalentă secvența push [ebp-8]; pop eax?

- a. mov ebp, eax; sub eax, 8
- b. mov eax, ebp; sub eax, 8
- c. mov eax, ebp-8
- d. mov eax, [ebp-8]
- e. mov [ebp-8], eax



Răspunsul dumneavoastră este corect.

The correct answer is: mov eax, [ebp-8]

Question 4
Correct
Mark 1.00 out of 1.00
 Flag question

Care dintre zonele de memorie din spațiul virtual de adrese al unui proces este dinamică cu alocare explicită?

- a. stack
- b. heap
- c. data
- d. code
- e. bss



Răspunsul dumneavoastră este corect.

The correct answer is: heap

[Finish review](#)

◀ 2CC: Recap 9: Moduri de adresare. Stiva

Jump to...



Laborator 6 ►

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 16 November - 22 November / 2CC: Recap 9: Moduri de adresare. Stiva

Started on	Thursday, 19 November 2020, 4:03 PM
State	Finished
Completed on	Thursday, 19 November 2020, 4:05 PM
Time taken	2 mins 8 secs
Marks	2.00/4.00
Grade	5.00 out of 10.00 (50%)

Quiz navigation



[Finish review](#)

Question 1

Correct

Mark 1.00 out of 1.00

 Flag question

Care dintre următoarele construcții în limbaj de asamblare este cea mai apropiată de construcția C int a = 1;?

- a. mov [eax], [ebx]
- b. mov eax, [ebx]
- c. mov [ax], 1
- d. mov [eax], 1
- e. mov [al], 1



Răspunsul dumneavoastră este corect.

The correct answer is: mov [eax], 1

Question 2

Partially correct

Mark 0.50 out of 1.00

 Flag question

Care dintre următoare construcții nu este validă? (alegeți 2 variante)

- a. mov [eax], [ebx+4]
- b. mov esp, [eip]
- c. mov esp, 5000
- d. dec [ebx]
- e. add eax, [ebp+ecx*4+12]
- f. sub esp, 30



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: mov esp, [eip], mov [eax], [ebx+4]

Question 3

Incorrect

Mark 0.00 out of 1.00

 Flag question

Care dintre următoarele înseamnă alocare pe stivă?

- a. mov esp, 16
- b. mov esp, ebp
- c. sub esp, 16
- d. add esp, 16
- e. pop eax



Răspunsul dumneavoastră este incorct.

The correct answer is: sub esp, 16

Question 4

Partially correct

Mark 0.50 out of 1.00

 Flag question

Care dintre următoarele construcții C folosesc stiva? (alegeți 2 variante)

- a. variabilele locale statice
- b. apelurile de funcții
- c. variabilele globale statice
- d. variabilele globale nestatice
- e. variabilele locale nestatice
- f. datele alocate dinamic



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: variabilele locale nestatice, apelurile de funcții

[Finish review](#)

[◀ Curs 12](#)

Jump to...



[2CC: Verificare: Stiva. Funcții ►](#)

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 9 November - 15 November / 2CC: Verificare: Moduri de adresare

Started on	Friday, 13 November 2020, 1:10 PM
State	Finished
Completed on	Friday, 13 November 2020, 1:11 PM
Time taken	59 secs
Marks	3.00/4.00
Grade	7.50 out of 10.00 (75%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Fie instrucțiunea mov eax, [var]. Ce se va găsi în registrul eax după realizarea instrucțiunii?

- a. adresa corespunzătoare etichetei var
- b. adresa variabilei var
- c. 4 octeți de la adresa indicată de var
- d. instrucțiunea va reuși doar dacă var este o etichetă pentru un vector
- e. 1 octet de la adresa indicată de var



Răspunsul dumneavoastră este corect.

The correct answer is: 4 octeți de la adresa indicată de var

Question 2
Correct
Mark 1.00 out of 1.00
Flag question

Fie instrucțiunea mov edx, var. Ce va conține registrul edx după realizarea acestei operații?

- a. instrucțiunea va reuși doar dacă var este definită cu resd
- b. 4 octeți de la adresa var
- c. adresa etichetei var
- d. instrucțiunea va reuși doar dacă var este definită cu dd
- e. 2 octeți de la adresa var



Răspunsul dumneavoastră este corect.

The correct answer is: adresa etichetei var

Question 3
Incorrect
Mark 0.00 out of 1.00
Flag question

Fie declarația var: dw 0x1122, 0x3344, 0x5566, 0x7788. Și instrucțiunea mov ecx, [var]. Ce valoare se va găsi în registrul ecx după realizarea acestei instrucțiuni?

- a. 0x33441122
- b. 0x11223344
- c. 0x1122
- d. 0x3344
- e. 0x22114433



Răspunsul dumneavoastră este incorrect.

The correct answer is: 0x33441122

Question 4
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele instrucțiuni este nevalidă?

- a. mov eax, [ebx]
- b. mov [ebx], eax
- c. mov [eax], [ebx]
- d. mov eax, ebx
- e. mov [eax + 4], ebx



Răspunsul dumneavoastră este corect.

The correct answer is: mov [eax], [ebx]

Finish review

◀ 2CC: Recap 8: Declararea datelor

Jump to...

Lucrare de curs 8: Moduri de adresare ►

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 9 November - 15 November / 2CC: Recap 8: Declararea datelor

Started on	Friday, 13 November 2020, 12:05 PM
State	Finished
Completed on	Friday, 13 November 2020, 12:06 PM
Time taken	1 min 40 secs
Marks	3.00/4.00
Grade	7.50 out of 10.00 (75%)

Quiz navigation

1	2	3	4
✓	✓	✓	✗

[Finish review](#)

Question 1
Correct
Mark 1.00 out of 1.00
[Flag question](#)

Care dintre următoarele operații NU vor cauza activarea overflow flag (OF)? (alegeți 2 variante)

a. adunarea unui număr pozitiv cu unul negativ
 b. adunarea a două numere negative
 c. scăderea dintr-un număr pozitiv un număr negativ
 d. scăderea dintr-un număr pozitiv un alt număr pozitiv
 e. adunarea a două numere pozitive

✓ adunarea a două numere negative

Răspunsul dumneavoastră este corect.

The correct answers are: adunarea unui număr pozitiv cu unul negativ, scăderea dintr-un număr pozitiv un alt număr pozitiv

Question 2
Correct
Mark 1.00 out of 1.00
[Flag question](#)

Care este echivalentul în limbaj de asamblare pentru construcția int v = 10;? (variabilă globală, int ocupă 32 de biți)

a. v: resb 10
 b. v: resd 10
 c. v: resw 10
 d. v: dw 10
 e. v: db 10
 f. v: dd 10

Răspunsul dumneavoastră este corect.

The correct answer is: v: dd 10

Question 3
Correct
Mark 1.00 out of 1.00
[Flag question](#)

Care este echivalentul în limbaj de asamblare pentru construcția int g;? (variabilă globală neinitializată, int ocupă 32 de biți)

a. g: db 1
 b. g: dd 1
 c. g: dw 1
 d. g: resb 1
 e. g: resd 1
 f. g: resw 1

Răspunsul dumneavoastră este corect.

The correct answer is: g: resd 1

Question 4
Incorrect
Mark 0.00 out of 1.00
[Flag question](#)

Cu ce este echivalentă, în limbaj de asamblare NASM, prezența cuvântului cheie static în fața unei variabile globale C?

a. cu declararea variabilei în secțiunea .data
 b. cu absența cuvântului cheie global din fața etichetei corespunzătoare în limbaj de asamblare
 c. cu declararea variabilei în secțiunea .bss
 d. cu absența cuvântului cheie local din fața etichetei corespunzătoare în limbaj de asamblare
 e. cu definirea etichetei corespunzătoare folosind equ (pentru constante)



Răspunsul dumneavoastră este incorrect.

The correct answer is: cu absența cuvântului cheie global din fața etichetei corespunzătoare în limbaj de asamblare

[Finish review](#)

[◀ 2CC: Recap 7: Setul de instrucțiuni](#)

Jump to... ▾

[2CC: Verificare: Moduri de adresare ►](#)



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 9 November - 15 November / 2CC: Recap 7: Setul de instrucțiuni

Started on Thursday, 12 November 2020, 4:13 PM

State Finished

Completed on Thursday, 12 November 2020, 4:14 PM

Time taken 58 secs

Marks 2.00/3.00

Grade **6.67** out of 10.00 (67%)

Question 1

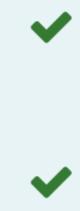
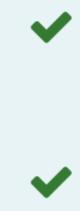
Correct

Mark 1.00 out of 1.00

Flag question

Care dintre următoarele instrucțiuni C generează instrucțiuni de salt înapoi în limbaj de asamblare? (alegeți 2 variante)

- a. for
- b. else
- c. while
- d. switch
- e. if
- f. typedef



Răspunsul dumneavoastră este corect.

The correct answers are: for, while

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

Care dintre următoarele rezultate ale unei operații de adunare a însemnat totodată și activarea flag-ului de semn (SF)? (alegeți 2 variante)

- a. 0x55
- b. 0xaa
- c. 0x80
- d. 0x4f
- e. 0x11
- f. 0x70



Răspunsul dumneavoastră este corect.

The correct answers are: 0x80, 0xaa

Question 3

Incorrect

Mark 0.00 out of 1.00

Flag question

Care dintre următoarele flag-uri nu este relevant pentru operații cu numere fără semn? (alegeți 2 variante)

- a. parity flag
- b. overflow flag
- c. zero flag
- d. carry flag
- e. sign flag



Răspunsul dumneavoastră este incorrect.

The correct answers are: sign flag, overflow flag

Finish review

◀ Laborator 5

Jump to...



2CC: Recap 8: Declararea datelor ►

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățământul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [2 November - 8 November](#) / [2CC: Recap 6: Setul de instrucțiuni](#)

Started on	Thursday, 5 November 2020, 4:02 PM
State	Finished
Completed on	Thursday, 5 November 2020, 4:04 PM
Time taken	1 min 38 secs
Marks	1.00/4.00
Grade	2.50 out of 10.00 (25%)

Question 1

Incorrect

Mark 0.00 out of 1.00

[Flag question](#)

Care dintre următoarele clase de instrucțiuni de transfer nu sunt valabile pe arhitectura x86? (alegeți 2 variante) Primul operand este destinația, al doilea operand este sursa.

- a. mov memorie, registru ✗
- b. mov memorie, registru ✗
- c. mov memorie, memorie
- d. mov registru, imediat
- e. mov registru, memorie
- f. mov imediat, registru

Răspunsul dumneavoastră este incorrect.

The correct answers are: mov imediat, registru, mov memorie, memorie

Quiz navigation

1	2	3	4
---	---	---	---

[Finish review](#)

Question 2

Incorrect

Mark 0.00 out of 1.00

[Flag question](#)

Care dintre următoarele instrucțiuni este pe x86 (CISC) echivalentul load (din RISC)? Primul operand este destinația, al doilea operand este sursa.

- a. mov registru, memorie
- b. mov memorie, registru
- c. mov registru, registru
- d. mov memorie, 0
- e. mov memorie, memorie
- f. mov registru, 0 ✗

Răspunsul dumneavoastră este incorrect.

The correct answer is: mov registru, memorie

Question 3

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Care dintre următoarele este echivalentă cu mov eax, 0?

- a. mov eax, ebx
- b. and eax, eax
- c. xor eax, eax ✓
- d. mov 0, eax
- e. test eax, eax
- f. inc eax

Răspunsul dumneavoastră este corect.

The correct answer is: xor eax, eax

Question 4

Incorrect

Mark 0.00 out of 1.00

[Flag question](#)

Ce instrucțiune este cel mai aproape ca efect de cmp eax, ebx?

- a. test eax, ebx
- b. and eax, ebx
- c. add eax, ebx
- d. sub eax, ebx
- e. xor eax, ebx ✗

Răspunsul dumneavoastră este incorrect.

The correct answer is: sub eax, ebx

[Finish review](#)[◀ Curs 7](#)

Jump to...

▼

[Curs 8-9 ►](#)

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 26 October - 1 November / 2CC: Recap 4: Reprezentarea datelor în sistemul de calcul

Started on	Thursday, 29 October 2020, 4:05 PM
State	Finished
Completed on	Thursday, 29 October 2020, 4:07 PM
Time taken	2 mins 17 secs
Marks	3.25/4.00
Grade	8.13 out of 10.00 (81%)

Question 1

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Fie numărul 0b11010011. Cât este bitul de semn?

- a. nu se poate spune
- b. 1
- c. 2
- d. 0
- e. are sens doar pentru numere în hexazecimal



Răspunsul dumneavoastră este corect.

The correct answer is: 1

Question 2

Partially correct

Mark 0.75 out of 1.00

[Flag question](#)

Asociați numere în hexazecimal cu forma în binar.

0x56	0b01010110	✓
0x81	0b10000001	✓
0xa3	0b01010110	✗
0x29	0b00101001	✓

Răspunsul dumneavoastră este parțial corect.

You have correctly selected 3.

The correct answer is: 0x56 → 0b01010110, 0x81 → 0b10000001, 0xa3 → 0b10100011, 0x29 → 0b00101001

Question 3

Partially correct

Mark 0.50 out of 1.00

[Flag question](#)

Care dintre următoarele sunt variante zecimale ale numărului hexazecimal 0x88? (alegeți cel mult 2 variante)

- a. 120
- b. -88
- c. -120
- d. 136
- e. -136
- f. 88



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: 136, -120

Question 4

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Care dintre următoarele sunt variante zecimale ale numărului hexazecimal 0x44? (alegeți cel mult 2 variante)

- a. -68
- b. 44
- c. 188
- d. -44
- e. 68
- f. -188



Răspunsul dumneavoastră este corect.

The correct answer is: 68

[Finish review](#)[◀ Laborator 3](#)

Jump to...

2CC: Verificare: Reprezentarea și interpretarea numerelor ►

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [26 October - 1 November](#) / [2CC: Verificare: Reprezentarea și interpretarea numerelor](#)

Started on	Thursday, 29 October 2020, 5:06 PM
State	Finished
Completed on	Thursday, 29 October 2020, 5:08 PM
Time taken	1 min 30 secs
Marks	4.00/4.00
Grade	10.00 out of 10.00 (100%)

Quiz navigation

1	2	3	4
✓	✓	✓	✓

[Finish review](#)

Question 1

Correct

Mark 1.00 out of 1.00

 Flag question

Care număr negativ e o interpretare posibilă pentru reprezentarea 0xaa?

- a. -70
- b. -170
- c. -86
- d. -76



Răspunsul dumneavoastră este corect.

The correct answer is: -86

Question 2

Correct

Mark 1.00 out of 1.00

 Flag question

Care număr negativ e o interpretare posibilă pentru reprezentarea 0xdd?

- a. -35
- b. -45
- c. -221
- d. -211



Răspunsul dumneavoastră este corect.

The correct answer is: -35

Question 3

Correct

Mark 1.00 out of 1.00

 Flag question

Care este reprezentarea hexazecimală a lui 200 (zecimal)?

- a. 0xc8
- b. 0xd0
- c. 0xd8
- d. 0xc0



Răspunsul dumneavoastră este corect.

The correct answer is: 0xc8

Question 4

Correct

Mark 1.00 out of 1.00

 Flag question

Care este reprezentarea hexazecimală a lui 169 (zecimal)?

- a. 0xd9
- b. 0xb9
- c. 0xa9
- d. 0xc9



Răspunsul dumneavoastră este corect.

The correct answer is: 0xa9

[Finish review](#)

[◀ 2CC: Recap 4: Reprezentarea datelor în sistemul de calcul](#)

Jump to...

[2CC: Recap 5: Reprezentarea datelor în sistemul de calcul \(partea a 2-a\) ►](#)

You are logged in as Andrei STAN ([Log out](#))

[03-ACS-L-A2-S1-IOCLA-CTI](#)

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: Limbajul de asamblare. Arhitectura sistemului de calcul

Started on	Friday, 16 October 2020, 1:16 PM
State	Finished
Completed on	Friday, 16 October 2020, 1:18 PM
Time taken	1 min 39 secs
Marks	3.00/4.00
Grade	7.50 out of 10.00 (75%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Unde se găsesc registrele?

- a. în procesor
- b. în executabil
- c. în memoria cache
- d. în loader
- e. în memoria RAM



Răspunsul dumneavoastră este corect.

The correct answer is: în procesor

Question 2
Incorrect
Mark 0.00 out of 1.00
Flag question

Ce înseamnă \$ (dollar) în NASM?

- a. sfârșitul liniei curente
- b. registrul de instrucțiuni
- c. începutul memoriei RAM
- d. adresa / poziția curentă curentă
- e. funcția main



Răspunsul dumneavoastră este incorrect.

The correct answer is: adresa / poziția curentă curentă

Question 3
Correct
Mark 1.00 out of 1.00
Flag question

Cum este conectat fizic procesorul la memoria principală (RAM)?

- a. registre
- b. magistrală
- c. I/O
- d. unitatea aritmetică și logică
- e. memorie cache



Răspunsul dumneavoastră este corect.

The correct answer is: magistrală

Question 4
Correct
Mark 1.00 out of 1.00
Flag question

Câte registre ne expune o arhitectură (ISA)? (ordin de mărime)

- a. 1000-2000
- b. egale cu dimensiunea memoriei RAM
- c. 10-20
- d. egale cu dimensiunea memoriei cache
- e. 100-200



Răspunsul dumneavoastră este corect.

The correct answer is: 10-20

Finish review

◀ 2CC: Recap 2: Programe și sistemul de calcul

Jump to...



Curs 4 ►

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: Recap 2: Programe și sistemul de calcul

Started on Friday, 16 October 2020, 12:03 PM

State Finished

Completed on Friday, 16 October 2020, 12:05 PM

Time taken 1 min 31 secs

Marks 3.50/4.00

Grade 8.75 out of 10.00 (88%)

Question 1

Correct

Mark 1.00 out of 1.00

Flag question

Fie construcția int v[10]; Ce NU este permis? (alegeți 2 variante)

- a. &v = 100;
- b. *v = 100;
- c. v = 100;
- d. *(v+4) = 100;
- e. v[9] = 100;



Răspunsul dumneavoastră este corect.

The correct answers are: &v = 100;; v = 100;

Question 2

Partially correct

Mark 0.50 out of 1.00

Flag question

Fie funcția main. Ce NU este permis? (alegeți 2 variante)

- a. printf("%p\n", &main);
- b. main = 100;
- c. *main = 100;
- d. printf("%p\n", *main);
- e. printf("%p\n", main);



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: *main = 100;; main = 100;

Question 3

Correct

Mark 1.00 out of 1.00

Flag question

Variabilele globale se găsesc în `section .data` ✓ .

Funcțiile se găsesc în `section .text` ✓ .

Răspunsul dumneavoastră este corect.

The correct answer is: Variabilele globale se găsesc în [section .data].

Funcțiile se găsesc în [section .text].

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

Care dintre următoarele instrucțiuni C NU folosește procesorul? (alegeți 2 variante)

- a. int v[100]; /* global variable */
- b. a++;
- c. my_func(); /* call my_func */
- d. int a;
- e. *a = 10;



Răspunsul dumneavoastră este corect.

The correct answers are: int a;; int v[100]; /* global variable */

Finish review

◀ 2CC: C what I see

Jump to...

2CC: Limbajul de asamblare. Arhitectura sistemului de calcul ►

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: C what I see

Started on Thursday, 15 October 2020, 5:36 PM

State Finished

Completed on Thursday, 15 October 2020, 5:38 PM

Time taken 1 min 47 secs

Marks 1.33/3.00

Grade 4.44 out of 10.00 (44%)

Question 1

Partially correct

Mark 0.33 out of 1.00

Flag question

Care dintre următoarele este sigur o adresă?

- a. int *p;
- b. 100
- c. *p
- d. funcția main
- e. #include
- f. #define a b
- g. &v;



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: int *p;, &v;, funcția main

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

Ce are o variabilă?

- a. adresă
- b. macro
- c. constructor
- d. valoare
- e. procesor
- f. dimensiune
- g. operație
- h. dușmani



Răspunsul dumneavoastră este corect.

The correct answers are: adresă, dimensiune, valoare

Question 3

Incorrect

Mark 0.00 out of 1.00

Flag question

Cu ce aproximăm cel mai bine memoria sistemului de calcul?



- a. funcție
- b. macro
- c. vector
- d. variabilă
- e. definiție
- f. union

Răspunsul dumneavoastră este incorct.

The correct answer is: vector

Finish review

◀ 2CC: Recap: Programe și sistemul de calcul

Jump to...



2CC: Recap 2: Programe și sistemul de calcul ►

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: Recap: Programe și sistemul de calcul

Started on	Thursday, 15 October 2020, 4:02 PM
State	Finished
Completed on	Thursday, 15 October 2020, 4:04 PM
Time taken	1 min 59 secs
Marks	3.50/4.00
Grade	8.75 out of 10.00 (88%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele sunt asociate 1-la-1?

- a. cod sursă - cod obiect
- b. cod mașină - limbaj de asamblare
- c. limbaj de asamblare - cod sursă
- d. limbaj de asamblare - cod sursă preprocesat



Răspunsul dumneavoastră este corect.

The correct answer is: cod mașină - limbaj de asamblare

Question 2
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele reprezintă cel mai bine interfața software - hardware?

- a. GCC
- b. DNS
- c. CPU
- d. RAM
- e. ISA



Răspunsul dumneavoastră este corect.

The correct answer is: ISA

Question 3
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele este cel mai low-level?

- a. bytecode-ul Java
- b. limbajul Java
- c. limbajul de asamblare
- d. limbajul C



Răspunsul dumneavoastră este corect.

The correct answer is: limbajul de asamblare

Question 4
Partially correct
Mark 0.50 out of 1.00
Flag question

Care dintre următoarele informații se referă DOAR la proces (nu și la program/executabil)? (alegeți 2 variante)

- a. cod mașină
- b. spațiu ocupat în memoria RAM
- c. cod ce poate fi dezasamblat
- d. timp pe procesor
- e. date



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: spațiu ocupat în memoria RAM, timp pe procesor

Finish review

◀ Laborator 1

Jump to...



2CC: C what I see ▶

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [25 October - 31 October](#) / [Lucrare de curs 1](#)

Started on Tuesday, 26 October 2021, 8:21 PM

State Finished

Completed on Tuesday, 26 October 2021, 8:28 PM

Time taken 7 mins 50 secs

Grade 10.00 out of 10.00 (100%)

Question 1

Correct

Mark 2.00 out of 2.00

Pe ce arhitectură poate fi rulat codul scris în C, fără a fi compilat în cod mașină?

- a. x86
- b. ARM
- c. MIPS
- d. Niciuna din variante



The correct answer is: Niciuna din variante

Question 2

Correct

Mark 2.00 out of 2.00

In timpul compilării unui program de la cod sursă la executabil, etapa de asamblare generează:

- a. Fișier ce conține cod în limbaj de asamblare
- b. Fișier ce conține cod C
- c. Fișier ce conține cod mașină
- d. Fișier ce conține cod C++



The correct answer is: Fișier ce conține cod mașină

Question 3

Correct

Mark 2.00 out of 2.00

Care dintre opțiunile de mai jos NU reprezintă un pas pe care îl face procesorul în momentul rulării unui program?

- a. Decodificarea instrucțiunilor
- b. Preluarea instrucțiunilor din memorie
- c. Executarea instrucțiunilor decodificate
- d. Tratarea erorilor de sintaxă apărute în programe



The correct answer is: Tratarea erorilor de sintaxă apărute în programe

Question 4

Correct

Mark 2.00 out of 2.00

Ce linie de cod generează, cu siguranță, eroare la compilare?

- a. `a = &10;`
- b. `a = **b;`
- c. `*a = *b;`
- d. `*a = 10;`



The correct answer is: `a = &10;`

Question 5

Correct

Mark 2.00 out of 2.00

Care este un avantaj al utilizării linkării dinamice în locul celei statice?

- a. portabilitatea executabilului obținut
- b. zona de cod a programului inițial nu mai este plasată în executabil
- c. executabilul obținut are o dimensiune mai mică
- d. datele programului inițial nu mai sunt plasate în executabil



The correct answer is: executabilul obținut are o dimensiune mai mică

[◀ Capitol 04: Reprezentarea numerelor](#)

Jump to...

[2CC: Recap: Curs 05: Reprezentarea numerelor ►](#)

Started on Tuesday, 16 November 2021, 9:37 PM**State** Finished**Completed on** Tuesday, 16 November 2021, 9:47 PM**Time taken** 10 mins 1 sec**Marks** 4.00/5.00**Grade** 8.00 out of 10.00 (80%)**Question 1**

Correct

Mark 1.00 out of 1.00

În C, care dintre următoarele condiții va fi adevărată dacă variabilele de tip int, nenule, x și y au semne diferite?

- a. $x \wedge y < 0$
- b. $x \& y > 0$
- c. $x \wedge y > 0$
- d. $(x \wedge y) \wedge (x \wedge y) > 0$



The correct answer is: $x \wedge y < 0$

Question 2

Correct

Mark 1.00 out of 1.00

Care afirmație este FALSĂ cu privire la instrucțiunea: test destination, source?

- a. Nu poate seta flaguri.
- b. Nu modifică destination.
- c. Nu modifică source.
- d. Efectuează și-logic între cei 2 operanzi.



The correct answer is: Nu poate seta flaguri.

Question 3

Correct

Mark 1.00 out of 1.00

Ce registru NU poate fi modificat direct (adică NU poate fi explicit modificat ca parte a unei instrucțiuni)?

- a. EAX
- b. EIP
- c. ESI
- d. DH



The correct answer is: EIP

Question 4

Correct

Mark 1.00 out of 1.00

Care dintre următoarele registre mai este numit și registru acumulator?

- a. EBX
- b. EDI
- c. EAX
- d. ESI



The correct answer is: EAX

Question 5

Incorrect

Mark 0.00 out of 1.00

Reprezentarea lui -1 în complement față de 2 este echivalentă cu:

- a. Același număr și în cazul reprezentării fără semn
- b. Cel mai mic număr dacă ar fi reprezentare fără semn
- c. 0
- d. Cel mai mare număr dacă ar fi reprezentare fără semn



The correct answer is: Cel mai mare număr dacă ar fi reprezentare fără semn

[◀ Capitol 06: Stiva](#)

Jump to...

[2CA: Recap: Curs 08: Interfața hardware-software x86 \(partea a 2-a\) ►](#)

Started on Tuesday, 14 December 2021, 9:58 PM**State** Finished**Completed on** Tuesday, 14 December 2021, 10:07 PM**Time taken** 9 mins 33 secs**Grade** 8.00 out of 10.00 (80%)**Question 1**

Correct

Mark 2.00 out of 2.00

Cu ce secvență în limbaj de asamblare este echivalentă secvența de cod C de mai jos? Considerăm că sistemul este pe 32 de biți și compilatorul nu reordonează variabilele locale.

```
void my_func(char *s) {  
    ...  
    int num_digits = strlen(s);  
    ...  
}
```

- a. sub esp, 4; push dword [ebp + 8]; call strlen; add esp, 4; mov [esp + 4], eax
- b. sub esp, 4; push dword [ebp + 4]; call strlen; add esp, 4; mov [ebp], eax
- c. add esp, 4; push dword [ebp + 8]; call strlen; sub esp, 4; mov [esp], eax
- d. sub esp, 4; push dword [ebp + 8]; call strlen; add esp, 4; mov [esp], eax



The correct answer is: sub esp, 4; push dword [ebp + 8]; call strlen; add esp, 4; mov [esp], eax

Question 2

Correct

Mark 2.00 out of 2.00

Care afirmație despre stiva unui program este FALSĂ pe arhitectura x86?

- a. destinația pentru "pop" poate fi un imediat
- b. poate fi folosită la stocarea datelor
- c. putem adăuga și scoate elemente de tip word și dword
- d. crește de la adrese mari la adrese mici



The correct answer is: destinația pentru "pop" poate fi un imediat

Question 3

Incorrect

Mark 0.00 out of 2.00

Trimitera parametrilor folosind stiva permite folosirea funcțiilor cu număr variabil de parametri.

- a. Fals
- b. Adevărat



The correct answer is: Adevărat

Question 4

Correct

Mark 2.00 out of 2.00

Fie semnătura de funcție "void my_func(short *a, int *b)". Cum pot fi accesate parametrii a, respectiv b pe un sistem pe 32 de biți?

- a. a: [ebp - 8]; b: [ebp - 16]
- b. a: [ebp + 10]; b: [ebp + 8]
- c. a: [ebp + 8]; b: [ebp + 10]
- d. a: [ebp + 8]; b: [ebp + 16]

✓ Răspunsul corect este, de fapt a:
[ebp + 8]; b: [ebp + 12]

The correct answers are: a: [ebp + 8]; b: [ebp + 16], a: [ebp - 8]; b: [ebp - 16], a: [ebp + 8]; b: [ebp + 10], a: [ebp + 10]; b: [ebp + 8]

Question 5

Correct

Mark 2.00 out of 2.00

Fie următoarele declarări de variabile:

```
var1 db 0x69
var2 dw 0xabcd
```

```
var3 db 0x12
```

și următoarea instrucțiune:

```
mov eax, [var1]
```

Ce valoare se va afla în registrul eax?

- a. 0x69cdab12
- b. 0x69abcd12
- c. 0x12abcd69
- d. 0x12cdab69



The correct answer is: 0x12abcd69

◀ 2CC: Recap: Curs 14: Interfața binară a funcțiilor

Jump to...

Capitol 09: Gestiona bufferelor ►

Started on Tuesday, 25 January 2022, 11:16 PM

State Finished

Completed on Tuesday, 25 January 2022, 11:20 PM

Time taken 3 mins 58 secs

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question 1

Correct

Mark 1.00 out of 1.00

Care dintre următoarele variante de răspuns este adevărată cu privire la directiva de asamblare global?

- a. Directiva global se folosește doar pentru variabile.
- b. Este obligatoriu să specificam tipul etichetei.
- c. Directiva global marchează o etichetă accesibilă și din alte module ale programului.
- d. Directiva global se folosește doar pentru nume de proceduri.



The correct answer is: Directiva global marchează o etichetă accesibilă și din alte module ale programului.

Question 2

Correct

Mark 1.00 out of 1.00

Reprezintă o metodă de protecție împotriva overflow:

- a. OOB (out of bounds)
- b. Canari
- c. Information leak
- d. Memory disclosure



The correct answer is: Canari

Question 3

Correct

Mark 1.00 out of 1.00

În cazul în care avem două fișiere: a.c și b.asm și dorim să obținem un executabil din cele două surse, ce utilitare vor fi folosite în acest proces?

- a. Compilator, asamblor și linker.
- b. Compilator și linker.
- c. Nu se poate obține executabil mixt.
- d. Asamblor și compilator.



The correct answer is: Compilator, asamblor și linker.

Question 4

Correct

Mark 1.00 out of 1.00

Care dintre următoarele reprezintă o unealtă de analiză dinamică?

- a. strings
- b. gdb
- c. readelf
- d. gcc



The correct answer is: gdb

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre următoarele poate fi considerat un DEZAVANTAJ al analizei dinamice?

- a. În general, acoperirea (coverage) este mai redusă decât în cazul analizei statice
- b. nu poate detecta vulnerabilități apărute la runtime (de exemplu segmentation fault)
- c. nu se poate realiza analiză dacă n-avem acces la codul sursă
- d. nu poate fi realizată pentru orice aplicație (inflexibilitate)



The correct answer is: În general, acoperirea (coverage) este mai redusă decât în cazul analizei statice

[◀ 2CC: Recap: Curs 17: Gestiunea bufferelor \(partea a 3-a\)](#)

Jump to...

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 11 January - 17 January / 2CC: Recap 15: Gestiunea bufferelor

Started on	Thursday, 14 January 2021, 4:05 PM
State	Finished
Completed on	Thursday, 14 January 2021, 4:07 PM
Time taken	1 min 49 secs
Marks	2.00/4.00
Grade	5.00 out of 10.00 (50%)

Question 1
Correct
Mark 1.00 out of 1.00

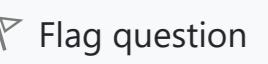

Pe un sistem pe 64 de biți, un buffer este la adresa rbp-72. Care este offsetul până la adresa de return?

- a. 88
- b. 76
- c. 72
- d. 80
- e. 64



Răspunsul dumneavoastră este corect.

The correct answer is: 80

Question 2
Correct
Mark 1.00 out of 1.00


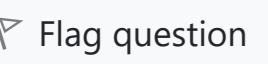
Pe un sistem pe 32 de biți folosim payloadul $64*\'A' + p32(hidden_address)$ ca să suprascriem adresa de return cu adresa funcției `hidden_address`. La ce adresă este bufferul?

- a. `ebp+64`
- b. `ebp-8`
- c. `ebp+4`
- d. `ebp-60`
- e. `ebp-64`



Răspunsul dumneavoastră este corect.

The correct answer is: `ebp-60`

Question 3
Incorrect
Mark 0.00 out of 1.00


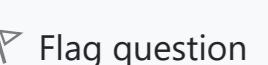
Un buffer este la adresa `ebp-40`, pe un sistem pe 32 de biți. Folosim payloadul $28*\'A' + p32(hidden_address)$. Ce vom suprascrie cu adresa `hidden_address`?

- a. un parametru al funcției
- b. fostul esp
- c. adresa de return
- d. fostul ebp
- e. o variabilă locală



Răspunsul dumneavoastră este incorrect.

The correct answer is: o variabilă locală

Question 4
Incorrect
Mark 0.00 out of 1.00


Un buffer este la adresa `ebp-32` pe un sistem pe 32 de biți. Dacă folosim funcția `fwrite()` pentru a scrie la standard output 56 de octeți de la adresa bufferului, ce se va întâmpla?

- a. vom face leak doar la adresa fostului `ebp`
- b. vom suprascrie doar fostul `ebp`
- c. vom face leak doar la adresa de return
- d. vom suprascrie fostul `ebp` și adresa de return
- e. vom face leak la adresa fostului `ebp` și adresa de return
- f. vom suprascrie adresa de return



Răspunsul dumneavoastră este incorrect.

The correct answer is: vom face leak la adresa fostului `ebp` și adresa de return

[Finish review](#)

◀ Curs 16-17

Jump to...



2CC: Recap 16: Optimizări ►

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 23 November - 29 November / 2CC: Recap 10: Funcții

Started on	Thursday, 26 November 2020, 4:06 PM
State	Finished
Completed on	Thursday, 26 November 2020, 4:08 PM
Time taken	1 min 28 secs
Marks	1.00/4.00
Grade	2.50 out of 10.00 (25%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Ce NU stocăm pe stivă?

- a. variabilele locale
- b. variabilele globale
- c. parametrii funcției
- d. adresa de return
- e. fostul frame pointer



Răspunsul dumneavoastră este corect.

The correct answer is: variabilele globale

Question 2
Incorrect
Mark 0.00 out of 1.00
Flag question

Cu ce este echivalentă instrucțiunea ret?

- a. mov eip, esp
- b. pop eip
- c. jmp eip
- d. push eip
- e. mov esp, eip



Răspunsul dumneavoastră este incorrect.

The correct answer is: pop eip

Question 3
Incorrect
Mark 0.00 out of 1.00
Flag question

Ce registru este frame pointer pe arhitectura x86?

- a. eflags
- b. ebp
- c. esi
- d. esp
- e. eip



Răspunsul dumneavoastră este incorrect.

The correct answer is: ebp

Question 4
Incorrect
Mark 0.00 out of 1.00
Flag question

Ce se găsește în vârful stivei la începutul execuției unei funcții?

- a. fostul frame pointer
- b. ultimul argument al funcției
- c. adresa următoarei instrucțiuni executate
- d. adresa de return
- e. primul argument al funcției



Răspunsul dumneavoastră este incorrect.

The correct answer is: adresa de return

Finish review

◀ Laborator 7

Jump to...



2CC: Recap 11: Funcții ►

You are logged in as Andrei STAN (Log out)

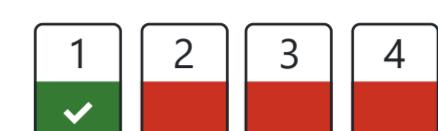
03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

Quiz navigation



Finish review



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 16 November - 22 November / 2CC: Verificare: Stiva. Funcții

Started on	Thursday, 19 November 2020, 5:04 PM
State	Finished
Completed on	Thursday, 19 November 2020, 5:06 PM
Time taken	2 mins 25 secs
Marks	2.00/4.00
Grade	5.00 out of 10.00 (50%)

Question 1
Incorrect
Mark 0.00 out of 1.00
Flag question

Ce indică stack pointer-ul?

- a. adresa curentă în program
- b. instrucțiunea următoare
- c. adresa funcției care urmează să fie apelată
- d. istoricul de apeluri de funcții
- e. valoarea cumulată a registrelor



Răspunsul dumneavoastră este incorrect.

The correct answer is: istoricul de apeluri de funcții

Question 2
Incorrect
Mark 0.00 out of 1.00
Flag question

Cum se numește fiecare element constituent din stiva unui program? Cum se numește fiecare element al stivei unui program?

- a. stack register
- b. stack frame
- c. back trace
- d. stack pointer
- e. stack heap



Răspunsul dumneavoastră este incorrect.

The correct answer is: stack frame

Question 3
Correct
Mark 1.00 out of 1.00
Flag question

Cu ce este echivalentă secvența push [ebp-8]; pop eax?

- a. mov ebp, eax; sub eax, 8
- b. mov eax, ebp; sub eax, 8
- c. mov eax, ebp-8
- d. mov eax, [ebp-8]
- e. mov [ebp-8], eax



Răspunsul dumneavoastră este corect.

The correct answer is: mov eax, [ebp-8]

Question 4
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre zonele de memorie din spațiul virtual de adrese al unui proces este dinamică cu alocare explicită?

- a. stack
- b. heap
- c. data
- d. code
- e. bss



Răspunsul dumneavoastră este corect.

The correct answer is: heap

Finish review

◀ 2CC: Recap 9: Moduri de adresare. Stiva

Jump to...

Laborator 6 ►

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 16 November - 22 November / 2CC: Recap 9: Moduri de adresare. Stiva

Started on	Thursday, 19 November 2020, 4:03 PM
State	Finished
Completed on	Thursday, 19 November 2020, 4:05 PM
Time taken	2 mins 8 secs
Marks	2.00/4.00
Grade	5.00 out of 10.00 (50%)

Quiz navigation



[Finish review](#)

Question 1

Correct

Mark 1.00 out of 1.00

 Flag question

Care dintre următoarele construcții în limbaj de asamblare este cea mai apropiată de construcția C int a = 1;?

- a. mov [eax], [ebx]
- b. mov eax, [ebx]
- c. mov [ax], 1
- d. mov [eax], 1
- e. mov [al], 1



Răspunsul dumneavoastră este corect.

The correct answer is: mov [eax], 1

Question 2

Partially correct

Mark 0.50 out of 1.00

 Flag question

Care dintre următoare construcții nu este validă? (alegeți 2 variante)

- a. mov [eax], [ebx+4]
- b. mov esp, [eip]
- c. mov esp, 5000
- d. dec [ebx]
- e. add eax, [ebp+ecx*4+12]
- f. sub esp, 30



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: mov esp, [eip], mov [eax], [ebx+4]

Question 3

Incorrect

Mark 0.00 out of 1.00

 Flag question

Care dintre următoarele înseamnă alocare pe stivă?

- a. mov esp, 16
- b. mov esp, ebp
- c. sub esp, 16
- d. add esp, 16
- e. pop eax



Răspunsul dumneavoastră este incorct.

The correct answer is: sub esp, 16

Question 4

Partially correct

Mark 0.50 out of 1.00

 Flag question

Care dintre următoarele construcții C folosesc stiva? (alegeți 2 variante)

- a. variabilele locale statice
- b. apelurile de funcții
- c. variabilele globale statice
- d. variabilele globale nestatice
- e. variabilele locale nestatice
- f. datele alocate dinamic



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: variabilele locale nestatice, apelurile de funcții

[Finish review](#)

[◀ Curs 12](#)

Jump to...



[2CC: Verificare: Stiva. Funcții ►](#)

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 9 November - 15 November / 2CC: Verificare: Moduri de adresare

Started on	Friday, 13 November 2020, 1:10 PM
State	Finished
Completed on	Friday, 13 November 2020, 1:11 PM
Time taken	59 secs
Marks	3.00/4.00
Grade	7.50 out of 10.00 (75%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Fie instrucțiunea mov eax, [var]. Ce se va găsi în registrul eax după realizarea instrucțiunii?

- a. adresa corespunzătoare etichetei var
- b. adresa variabilei var
- c. 4 octeți de la adresa indicată de var
- d. instrucțiunea va reuși doar dacă var este o etichetă pentru un vector
- e. 1 octet de la adresa indicată de var



Răspunsul dumneavoastră este corect.

The correct answer is: 4 octeți de la adresa indicată de var

Question 2
Correct
Mark 1.00 out of 1.00
Flag question

Fie instrucțiunea mov edx, var. Ce va conține registrul edx după realizarea acestei operații?

- a. instrucțiunea va reuși doar dacă var este definită cu resd
- b. 4 octeți de la adresa var
- c. adresa etichetei var
- d. instrucțiunea va reuși doar dacă var este definită cu dd
- e. 2 octeți de la adresa var



Răspunsul dumneavoastră este corect.

The correct answer is: adresa etichetei var

Question 3
Incorrect
Mark 0.00 out of 1.00
Flag question

Fie declarația var: dw 0x1122, 0x3344, 0x5566, 0x7788. Și instrucțiunea mov ecx, [var]. Ce valoare se va găsi în registrul ecx după realizarea acestei instrucțiuni?

- a. 0x33441122
- b. 0x11223344
- c. 0x1122
- d. 0x3344
- e. 0x22114433



Răspunsul dumneavoastră este incorrect.

The correct answer is: 0x33441122

Question 4
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele instrucțiuni este nevalidă?

- a. mov eax, [ebx]
- b. mov [ebx], eax
- c. mov [eax], [ebx]
- d. mov eax, ebx
- e. mov [eax + 4], ebx



Răspunsul dumneavoastră este corect.

The correct answer is: mov [eax], [ebx]

Finish review

◀ 2CC: Recap 8: Declararea datelor

Jump to...

Lucrare de curs 8: Moduri de adresare ►

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 9 November - 15 November / 2CC: Recap 8: Declararea datelor

Started on	Friday, 13 November 2020, 12:05 PM
State	Finished
Completed on	Friday, 13 November 2020, 12:06 PM
Time taken	1 min 40 secs
Marks	3.00/4.00
Grade	7.50 out of 10.00 (75%)

Quiz navigation

1	2	3	4
✓	✓	✓	✗

[Finish review](#)

Question 1

Correct

Mark 1.00 out of 1.00

 Flag question

Care dintre următoarele operații NU vor cauza activarea overflow flag (OF)? (alegeți 2 variante)

- a. adunarea unui număr pozitiv cu unul negativ
- b. adunarea a două numere negative
- c. scăderea dintr-un număr pozitiv un număr negativ
- d. scăderea dintr-un număr pozitiv un alt număr pozitiv
- e. adunarea a două numere pozitive

✓ adunarea a două numere negative

Răspunsul dumneavoastră este corect.

The correct answers are: adunarea unui număr pozitiv cu unul negativ, scăderea dintr-un număr pozitiv un alt număr pozitiv

Question 2

Correct

Mark 1.00 out of 1.00

 Flag question

Care este echivalentul în limbaj de asamblare pentru construcția int v = 10;? (variabilă globală, int ocupă 32 de biți)

- a. v: resb 10
- b. v: resd 10
- c. v: resw 10
- d. v: dw 10
- e. v: db 10
- f. v: dd 10

✓

Răspunsul dumneavoastră este corect.

The correct answer is: v: dd 10

Question 3

Correct

Mark 1.00 out of 1.00

 Flag question

Care este echivalentul în limbaj de asamblare pentru construcția int g;? (variabilă globală neinitializată, int ocupă 32 de biți)

- a. g: db 1
- b. g: dd 1
- c. g: dw 1
- d. g: resb 1
- e. g: resd 1
- f. g: resw 1

✓

Răspunsul dumneavoastră este corect.

The correct answer is: g: resd 1

Question 4

Incorrect

Mark 0.00 out of 1.00

 Flag question

Cu ce este echivalentă, în limbaj de asamblare NASM, prezența cuvântului cheie static în fața unei variabile globale C?

- a. cu declararea variabilei în secțiunea .data
- b. cu absența cuvântului cheie global din fața etichetei corespunzătoare în limbaj de asamblare
- c. cu declararea variabilei în secțiunea .bss
- d. cu absența cuvântului cheie local din fața etichetei corespunzătoare în limbaj de asamblare
- e. cu definirea etichetei corespunzătoare folosind equ (pentru constante)

✗

Răspunsul dumneavoastră este incorrect.

The correct answer is: cu absența cuvântului cheie global din fața etichetei corespunzătoare în limbaj de asamblare

[Finish review](#)

[◀ 2CC: Recap 7: Setul de instrucțiuni](#)

Jump to...

[2CC: Verificare: Moduri de adresare ►](#)

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 9 November - 15 November / 2CC: Recap 7: Setul de instrucțiuni

Started on Thursday, 12 November 2020, 4:13 PM

State Finished

Completed on Thursday, 12 November 2020, 4:14 PM

Time taken 58 secs

Marks 2.00/3.00

Grade **6.67** out of 10.00 (67%)

Question 1

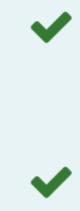
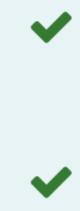
Correct

Mark 1.00 out of 1.00

Flag question

Care dintre următoarele instrucțiuni C generează instrucțiuni de salt înapoi în limbaj de asamblare? (alegeți 2 variante)

- a. for
- b. else
- c. while
- d. switch
- e. if
- f. typedef



Răspunsul dumneavoastră este corect.

The correct answers are: for, while

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

Care dintre următoarele rezultate ale unei operații de adunare a însemnat totodată și activarea flag-ului de semn (SF)? (alegeți 2 variante)

- a. 0x55
- b. 0xaa
- c. 0x80
- d. 0x4f
- e. 0x11
- f. 0x70



Răspunsul dumneavoastră este corect.

The correct answers are: 0x80, 0xaa

Question 3

Incorrect

Mark 0.00 out of 1.00

Flag question

Care dintre următoarele flag-uri nu este relevant pentru operații cu numere fără semn? (alegeți 2 variante)

- a. parity flag
- b. overflow flag
- c. zero flag
- d. carry flag
- e. sign flag



Răspunsul dumneavoastră este incorrect.

The correct answers are: sign flag, overflow flag

Finish review

◀ Laborator 5

Jump to...

Finish review

2CC: Recap 8: Declararea datelor ►

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățământul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [2 November - 8 November](#) / [2CC: Recap 6: Setul de instrucțiuni](#)

Started on	Thursday, 5 November 2020, 4:02 PM
State	Finished
Completed on	Thursday, 5 November 2020, 4:04 PM
Time taken	1 min 38 secs
Marks	1.00/4.00
Grade	2.50 out of 10.00 (25%)

Question 1

Incorrect

Mark 0.00 out of 1.00

 Flag question

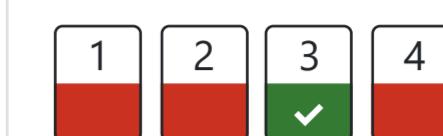
Care dintre următoarele clase de instrucțiuni de transfer nu sunt valabile pe arhitectura x86? (alegeți 2 variante) Primul operand este destinația, al doilea operand este sursa.

- a. mov memorie, registru ✗
- b. mov memorie, registru ✗
- c. mov memorie, memorie
- d. mov registru, imediat
- e. mov registru, memorie
- f. mov imediat, registru

Răspunsul dumneavoastră este incorrect.

The correct answers are: mov imediat, registru, mov memorie, memorie

Quiz navigation

[Finish review](#)**Question 2**

Incorrect

Mark 0.00 out of 1.00

 Flag question

Care dintre următoarele instrucțiuni este pe x86 (CISC) echivalentul load (din RISC)? Primul operand este destinația, al doilea operand este sursa.

- a. mov registru, memorie
- b. mov memorie, registru
- c. mov registru, registru
- d. mov memorie, 0
- e. mov memorie, memorie
- f. mov registru, 0 ✗

Răspunsul dumneavoastră este incorrect.

The correct answer is: mov registru, memorie

Question 3

Correct

Mark 1.00 out of 1.00

 Flag question

Care dintre următoarele este echivalentă cu mov eax, 0?

- a. mov eax, ebx
- b. and eax, eax
- c. xor eax, eax ✓
- d. mov 0, eax
- e. test eax, eax
- f. inc eax

Răspunsul dumneavoastră este corect.

The correct answer is: xor eax, eax

Question 4

Incorrect

Mark 0.00 out of 1.00

 Flag question

Ce instrucțiune este cel mai aproape ca efect de cmp eax, ebx?

- a. test eax, ebx
- b. and eax, ebx
- c. add eax, ebx
- d. sub eax, ebx
- e. xor eax, ebx ✗

Răspunsul dumneavoastră este incorrect.

The correct answer is: sub eax, ebx

[Finish review](#)[◀ Curs 7](#)

Jump to...

▼

[Curs 8-9 ►](#)

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 26 October - 1 November / 2CC: Recap 4: Reprezentarea datelor în sistemul de calcul

Started on	Thursday, 29 October 2020, 4:05 PM
State	Finished
Completed on	Thursday, 29 October 2020, 4:07 PM
Time taken	2 mins 17 secs
Marks	3.25/4.00
Grade	8.13 out of 10.00 (81%)

Question 1

Correct

Mark 1.00 out of 1.00

 Flag question

Fie numărul 0b11010011. Cât este bitul de semn?

- a. nu se poate spune
- b. 1
- c. 2
- d. 0
- e. are sens doar pentru numere în hexazecimal



Răspunsul dumneavoastră este corect.

The correct answer is: 1

Question 2

Partially correct

Mark 0.75 out of 1.00

 Flag question

Asociați numere în hexazecimal cu forma în binar.

0x56	0b01010110	▼	
0x81	0b10000001	▼	
0xa3	0b01010110	▼	
0x29	0b00101001	▼	

Răspunsul dumneavoastră este parțial corect.

You have correctly selected 3.

The correct answer is: 0x56 → 0b01010110, 0x81 → 0b10000001, 0xa3 → 0b10100011, 0x29 → 0b00101001

Question 3

Partially correct

Mark 0.50 out of 1.00

 Flag question

Care dintre următoarele sunt variante zecimale ale numărului hexazecimal 0x88? (alegeți cel mult 2 variante)

- a. 120
- b. -88
- c. -120
- d. 136
- e. -136
- f. 88



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: 136, -120

Question 4

Correct

Mark 1.00 out of 1.00

 Flag question

Care dintre următoarele sunt variante zecimale ale numărului hexazecimal 0x44? (alegeți cel mult 2 variante)

- a. -68
- b. 44
- c. 188
- d. -44
- e. 68
- f. -188



Răspunsul dumneavoastră este corect.

The correct answer is: 68

[Finish review](#)[◀ Laborator 3](#)[Jump to...](#)[2CC: Verificare: Reprezentarea și interpretarea numerelor ►](#)

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [26 October - 1 November](#) / [2CC: Verificare: Reprezentarea și interpretarea numerelor](#)

Started on	Thursday, 29 October 2020, 5:06 PM
State	Finished
Completed on	Thursday, 29 October 2020, 5:08 PM
Time taken	1 min 30 secs
Marks	4.00/4.00
Grade	10.00 out of 10.00 (100%)

Quiz navigation

1	2	3	4
✓	✓	✓	✓

[Finish review](#)

Question 1

Correct

Mark 1.00 out of 1.00

 Flag question

Care număr negativ e o interpretare posibilă pentru reprezentarea 0xaa?

- a. -70
- b. -170
- c. -86
- d. -76



Răspunsul dumneavoastră este corect.

The correct answer is: -86

Question 2

Correct

Mark 1.00 out of 1.00

 Flag question

Care număr negativ e o interpretare posibilă pentru reprezentarea 0xdd?

- a. -35
- b. -45
- c. -221
- d. -211



Răspunsul dumneavoastră este corect.

The correct answer is: -35

Question 3

Correct

Mark 1.00 out of 1.00

 Flag question

Care este reprezentarea hexazecimală a lui 200 (zecimal)?

- a. 0xc8
- b. 0xd0
- c. 0xd8
- d. 0xc0



Răspunsul dumneavoastră este corect.

The correct answer is: 0xc8

Question 4

Correct

Mark 1.00 out of 1.00

 Flag question

Care este reprezentarea hexazecimală a lui 169 (zecimal)?

- a. 0xd9
- b. 0xb9
- c. 0xa9
- d. 0xc9



Răspunsul dumneavoastră este corect.

The correct answer is: 0xa9

[Finish review](#)

[◀ 2CC: Recap 4: Reprezentarea datelor în sistemul de calcul](#)

Jump to...

[2CC: Recap 5: Reprezentarea datelor în sistemul de calcul \(partea a 2-a\) ►](#)

You are logged in as Andrei STAN ([Log out](#))

[03-ACS-L-A2-S1-IOCLA-CTI](#)

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: Limbajul de asamblare. Arhitectura sistemului de calcul

Started on	Friday, 16 October 2020, 1:16 PM
State	Finished
Completed on	Friday, 16 October 2020, 1:18 PM
Time taken	1 min 39 secs
Marks	3.00/4.00
Grade	7.50 out of 10.00 (75%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Unde se găsesc registrele?

- a. în procesor
- b. în executabil
- c. în memoria cache
- d. în loader
- e. în memoria RAM



Răspunsul dumneavoastră este corect.

The correct answer is: în procesor

Question 2
Incorrect
Mark 0.00 out of 1.00
Flag question

Ce înseamnă \$ (dollar) în NASM?

- a. sfârșitul liniei curente
- b. registrul de instrucțiuni
- c. începutul memoriei RAM
- d. adresa / poziția curentă curentă
- e. funcția main



Răspunsul dumneavoastră este incorrect.

The correct answer is: adresa / poziția curentă curentă

Question 3
Correct
Mark 1.00 out of 1.00
Flag question

Cum este conectat fizic procesorul la memoria principală (RAM)?

- a. registre
- b. magistrală
- c. I/O
- d. unitatea aritmetică și logică
- e. memorie cache



Răspunsul dumneavoastră este corect.

The correct answer is: magistrală

Question 4
Correct
Mark 1.00 out of 1.00
Flag question

Câte registre ne expune o arhitectură (ISA)? (ordin de mărime)

- a. 1000-2000
- b. egale cu dimensiunea memoriei RAM
- c. 10-20
- d. egale cu dimensiunea memoriei cache
- e. 100-200



Răspunsul dumneavoastră este corect.

The correct answer is: 10-20

Finish review

◀ 2CC: Recap 2: Programe și sistemul de calcul

Jump to...



Curs 4 ►

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru invatamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: Recap 2: Programe și sistemul de calcul

Started on Friday, 16 October 2020, 12:03 PM

State Finished

Completed on Friday, 16 October 2020, 12:05 PM

Time taken 1 min 31 secs

Marks 3.50/4.00

Grade 8.75 out of 10.00 (88%)

Question 1

Correct

Mark 1.00 out of 1.00

Flag question

Fie construcția int v[10]; Ce NU este permis? (alegeți 2 variante)

- a. &v = 100;
- b. *v = 100;
- c. v = 100;
- d. *(v+4) = 100;
- e. v[9] = 100;



Răspunsul dumneavoastră este corect.

The correct answers are: &v = 100;; v = 100;

Question 2

Partially correct

Mark 0.50 out of 1.00

Flag question

Fie funcția main. Ce NU este permis? (alegeți 2 variante)

- a. printf("%p\n", &main);
- b. main = 100;
- c. *main = 100;
- d. printf("%p\n", *main);
- e. printf("%p\n", main);



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: *main = 100;; main = 100;

Question 3

Correct

Mark 1.00 out of 1.00

Flag question

Variabilele globale se găsesc în section .data ✓ .

Funcțiile se găsesc în section .text ✓ .

Răspunsul dumneavoastră este corect.

The correct answer is: Variabilele globale se găsesc în [section .data].

Funcțiile se găsesc în [section .text].

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

Care dintre următoarele instrucțiuni C NU folosește procesorul? (alegeți 2 variante)

- a. int v[100]; /* global variable */
- b. a++;
- c. my_func(); /* call my_func */
- d. int a;
- e. *a = 10;



Răspunsul dumneavoastră este corect.

The correct answers are: int a;; int v[100]; /* global variable */

Finish review

◀ 2CC: C what I see

Jump to...

2CC: Limbajul de asamblare. Arhitectura sistemului de calcul ►

You are logged in as Andrei STAN ([Log out](#))

03-ACS-L-A2-S1-IOCLA-CTI

[Data retention summary](#)

[Get the mobile app](#)

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.



03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: C what I see

Started on Thursday, 15 October 2020, 5:36 PM

State Finished

Completed on Thursday, 15 October 2020, 5:38 PM

Time taken 1 min 47 secs

Marks 1.33/3.00

Grade 4.44 out of 10.00 (44%)

Question 1

Partially correct

Mark 0.33 out of 1.00

Flag question

Care dintre următoarele este sigur o adresă?

- a. int *p;
- b. 100
- c. *p
- d. funcția main
- e. #include
- f. #define a b
- g. &v;



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: int *p;, &v;, funcția main

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

Ce are o variabilă?

- a. adresă
- b. macro
- c. constructor
- d. valoare
- e. procesor
- f. dimensiune
- g. operație
- h. dușmani



Răspunsul dumneavoastră este corect.

The correct answers are: adresă, dimensiune, valoare

Question 3

Incorrect

Mark 0.00 out of 1.00

Flag question

Cu ce aproximăm cel mai bine memoria sistemului de calcul?



- a. funcție
- b. macro
- c. vector
- d. variabilă
- e. definiție
- f. union

Răspunsul dumneavoastră este incorct.

The correct answer is: vector

Finish review

◀ 2CC: Recap: Programe și sistemul de calcul

Jump to...



2CC: Recap 2: Programe și sistemul de calcul ►

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

03-ACS-L-A2-S1: Introducere în organizarea calculatoarelor și limbaj de asamblare (Seria CA, CB, CC, CD - 2020)

Dashboard / My courses / 03-ACS-L-A2-S1-IOCLA-CTI / 12 October - 18 October / 2CC: Recap: Programe și sistemul de calcul

Started on	Thursday, 15 October 2020, 4:02 PM
State	Finished
Completed on	Thursday, 15 October 2020, 4:04 PM
Time taken	1 min 59 secs
Marks	3.50/4.00
Grade	8.75 out of 10.00 (88%)

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele sunt asociate 1-la-1?

- a. cod sursă - cod obiect
- b. cod mașină - limbaj de asamblare
- c. limbaj de asamblare - cod sursă
- d. limbaj de asamblare - cod sursă preprocesat



Răspunsul dumneavoastră este corect.

The correct answer is: cod mașină - limbaj de asamblare

Question 2
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele reprezintă cel mai bine interfața software - hardware?

- a. GCC
- b. DNS
- c. CPU
- d. RAM
- e. ISA



Răspunsul dumneavoastră este corect.

The correct answer is: ISA

Question 3
Correct
Mark 1.00 out of 1.00
Flag question

Care dintre următoarele este cel mai low-level?

- a. bytecode-ul Java
- b. limbajul Java
- c. limbajul de asamblare
- d. limbajul C



Răspunsul dumneavoastră este corect.

The correct answer is: limbajul de asamblare

Question 4
Partially correct
Mark 0.50 out of 1.00
Flag question

Care dintre următoarele informații se referă DOAR la proces (nu și la program/executabil)? (alegeți 2 variante)

- a. cod mașină
- b. spațiu ocupat în memoria RAM
- c. cod ce poate fi dezasamblat
- d. timp pe procesor
- e. date



Răspunsul dumneavoastră este parțial corect.

You have correctly selected 1.

The correct answers are: spațiu ocupat în memoria RAM, timp pe procesor

Finish review

◀ Laborator 1

Jump to...



2CC: C what I see ▶

You are logged in as Andrei STAN (Log out)

03-ACS-L-A2-S1-IOCLA-CTI

Data retention summary

Get the mobile app

Acest site este hostat pe platforma hardware achiziționată din proiectul nr. 154/323 cod SMIS - 4428, "Platforma de e-learning și curricula e-content pentru învățamantul superior tehnic". Pentru mai multe detalii vezi <http://www.curs.pub.ro>.

Started on Friday, 9 October 2020, 6:56 PM**State** Finished**Completed on** Friday, 9 October 2020, 6:57 PM**Time taken** 44 secs**Marks** 5.00/5.00**Grade** **10.00** out of 10.00 (**100%**)**Question 1**

Correct

Mark 1.00 out of 1.00

Care sunt etapele compilării (în ordine) a unui program de la cod sursă la executabil?

- a. Preprocesare, asamblare, compilare, linkare
- b. Compilare, preprocesare, linkare, asamblare
- c. Preprocesare, compilare, asamblare, linkare
- d. Preprocesare, compilare, linkare, asamblare



The correct answer is: Preprocesare, compilare, asamblare, linkare

Question 2

Correct

Mark 1.00 out of 1.00

Care dintre opțiunile de mai jos NU reprezintă un avantaj al folosirii unui compilator?

- a. Oferă portabilitate între diferite arhitecturi de procesor
- b. Generează automat cod sursă din descrierea în limbaj natural
- c. Optimizează codul scris în limbaj de nivel înalt în măsura în care se poate
- d. Ascunde complexitatea codului mașină



The correct answer is: Generează automat cod sursă din descrierea în limbaj natural

Question 3

Correct

Mark 1.00 out of 1.00

Care NU este un avantaj al folosirii unui limbaj de nivel înalt în locul folosirii unuia de nivel jos?

- a. Portabilitatea
- b. Lucrul apropiat de hardware
- c. Ușurința citirii codului
- d. Dezvoltarea mai rapidă



The correct answer is: Lucrul apropiat de hardware

Question 4

Correct

Mark 1.00 out of 1.00

Care dintre variantele de mai jos NU este un exemplu de arhitectură de procesor?

- a. PowerPC
- b. macOS
- c. ARM
- d. MIPS



The correct answer is: macOS

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre opțiunile de mai jos NU reprezintă un pas pe care îl face procesorul în momentul rulării unui program?

- a. Tratarea erorilor de sintaxă apărute în programe
- b. Preluarea instrucțiunilor din memorie
- c. Decodificarea instrucțiunilor
- d. Executarea instrucțiunilor decodificate



The correct answer is: Tratarea erorilor de sintaxă apărute în programe

[◀ Tema 4](#)

Jump to...

[Curs 0 ►](#)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [12 October - 18 October](#) / [Lucrare de curs 2: ASC + Arhitectura x86](#)

Started on Friday, 16 October 2020, 4:15 PM

State Finished

Completed on Friday, 16 October 2020, 4:17 PM

Time taken 2 mins 3 secs

Marks 3.00/5.00

Grade **6.00** out of 10.00 (**60%**)

Question 1

Correct

Mark 1.00 out of 1.00

Care sunt pașii pentru crearea unui fișier executabil plecând de la un cod sursă?

- a. compilare, preprocesare, asamblare, link-editare
- b. preprocesare, compilare, link-editare, asamblare
- c. preprocesare, asamblare, link-editare, compilare
- d. preprocesare, compilare, asamblare, link-editare



The correct answer is: preprocesare, compilare, asamblare, link-editare

Question 2

Incorrect

Mark 0.00 out of 1.00

Care este relația dintre latența memoriilor și capacitatea lor?

- a. cu cât latența este mai mica, capacitatea este mai mica
- b. cu cât latența este mai mare, capacitatea este mai mare
- c. cu cât latența este mai mica, capacitatea este mai mare
- d. sunt independente



The correct answer is: cu cât latența este mai mica, capacitatea este mai mare

Question 3

Incorrect

Mark 0.00 out of 1.00

Prin intermediul cărei componente se realizează stocarea de date?

- a. procesor
- b. I/O
- c. CPU
- d. RAM



The correct answer is: RAM

Question 4

Correct

Mark 1.00 out of 1.00

Alegeți afirmația corectă în ceea ce privește limbajele de nivel înalt:

- a. programele nu sunt portabile
- b. dezvoltarea este mai lentă
- c. dezvoltarea este mai rapidă
- d. mențenanța este dificilă



The correct answer is: dezvoltarea este mai rapidă

Question 5

Correct

Mark 1.00 out of 1.00

Ce se întâmplă dacă se încearcă accesarea unei zone de memorie, printr-o adresă nevalidă (care nu a fost alocată în prealabil) ?

- a. programul funcționează corect
- b. eroare la compilare
- c. segmentation fault
- d. adresa respectivă devine validă



The correct answer is: segmentation fault

[◀ Curs 3](#)[Jump to...](#)[Laborator 1 ►](#)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [19 October - 25 October](#) / [Lucrare de curs 3: Arhitectura x86](#)

Started on Monday, 19 October 2020, 3:27 PM

State Finished

Completed on Monday, 19 October 2020, 3:32 PM

Time taken 4 mins 31 secs

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele zone din memoria unui program crește "în jos" (se folosește de la adrese mari la adrese mici)?

- a. Heap-ul
- b. Stiva ✓
- c. Segmentul de date
- d. Segmentul BSS

The correct answer is: Stiva

Question **2**

Correct

Mark 1.00 out of 1.00

Ce registru NU poate fi modificat direct (adică NU poate fi explicit modificat ca parte a unei instrucțiuni)?

- a. ESI
- b. DH
- c. EAX
- d. EIP ✓

The correct answer is: EIP

Question **3**

Correct

Mark 1.00 out of 1.00

După execuția unei instrucțiuni arbitrară, ce registru va fi întotdeauna afectat?

- a. ESP
- b. EIP ✓
- c. EDI
- d. EBP

The correct answer is: EIP

Question 4

Correct

Mark 1.00 out of 1.00

Care dintre următoarele registre este considerat un registru pointer (ține adesea o adresă)?

- a. EFLAGS
- b. ECX
- c. BH
- d. EBP



The correct answer is: EBP

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre următoarele registre mai este numit și registru acumulator?

- a. EDI
- b. ESI
- c. EBX
- d. EAX



The correct answer is: EAX

[◀ 2CC: Recap 3: Arhitectura sistemului de calcul](#)

Jump to...

[Lucrare de curs 4: Reprezentarea Datelor în Sistemele de Calcul ►](#)

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [19 October - 25 October](#)

/ [Lucrare de curs 4: Reprezentarea Datelor în Sistemele de Calcul](#)

Started on Friday, 23 October 2020, 6:04 PM

State Finished

Completed on Friday, 23 October 2020, 6:07 PM

Time taken 3 mins 1 sec

Marks 4.00/5.00

Grade **8.00** out of 10.00 (**80%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care operație este echivalentă cu împărțirea numărului n cu 16?

- a. $n \gg 4$
- b. $n \gg 16$
- c. $n << 16$
- d. $n << 4$



The correct answer is: $n \gg 4$

Question **2**

Correct

Mark 1.00 out of 1.00

Fie variabila unsigned char n = 255. Ce valoare va avea ea dacă îi facem cast la signed char?

- a. eroare de compilare
- b. -1
- c. 255
- d. 0



The correct answer is: -1

Question 3

Correct

Mark 1.00 out of 1.00

Pe câți biți se reprezintă o variabilă de tip long long?

- a. 16 biți
- b. 64 de biți
- c. 32 de biți
- d. 8 biți



The correct answer is: 64 de biți

Question 4

Correct

Mark 1.00 out of 1.00

Cum putem verifica că a și b diferă printr-un singur bit?

- a. $a \wedge b > 0$
- b. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \& b$
- c. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$
- d. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \mid b$



The correct answer is: $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$

Question 5

Incorrect

Mark 0.00 out of 1.00

Care dintre următoarele condiții este necesară ca numerele nenule x și y să aibă semne diferite?

- a. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \& b$
- b. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$
- c. $a \wedge b > 0$
- d. $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \mid b$



The correct answer is: $x \&\& (!(\text{x} \& (\text{x} - 1)))$, unde $\text{x} = a \wedge b$

[◀ Lucrare de curs 3: Arhitectura x86](#)

Jump to...

Laborator 3 ►

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [26 October - 1 November](#)

/ [Lucrare de curs 5: Reprezentarea Datelor în Sistemele de Calcul - C2](#)

Started on Friday, 30 October 2020, 6:53 PM

State Finished

Completed on Friday, 30 October 2020, 6:57 PM

Time taken 3 mins 32 secs

Marks 4.00/5.00

Grade **8.00** out of 10.00 (**80%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care este domeniul de valori ce poate fi stocat de o variabilă de tip short?

- a. [-2¹⁶, 2¹⁶]
- b. [0, 2¹⁶]
- c. [-16, 16]
- d. [-2¹⁵, 2¹⁵ - 1]



The correct answer is: [-2¹⁵, 2¹⁵ - 1]

Question **2**

Correct

Mark 1.00 out of 1.00

Cât înseamnă 0xFF în reprezentarea fară semn?

- a. -1
- b. 255
- c. -127
- d. -255



The correct answer is: 255

Question 3

Correct

Mark 1.00 out of 1.00

Pe câți biți se reprezintă un vector de tip unsigned char cu 3 elemente?

- a. 24 biți
- b. 48 de biți
- c. 8 biți
- d. 36 de biți



The correct answer is: 24 biți

Question 4

Correct

Mark 1.00 out of 1.00

Fie variabila char n = -100. Ce valoare va avea ea dacă îi facem cast la unsigned char?

- a. 227
- b. eroare de compilare
- c. 124
- d. 156



The correct answer is: 156

Question 5

Incorrect

Mark 0.00 out of 1.00

Fie declarația char s[] = "ANA"; Știind că A = 0x41, N=0x4E în ce se termina sirul declarat s?

- a. 0x1
- b. 0x41
- c. 0x4E
- d. 0x00



The correct answer is: 0x00

[◀ 2CC: Recap 5: Reprezentarea datelor în sistemul de calcul \(partea a 2-a\)](#)

Jump to...

Curs 7 ►

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [2 November - 8 November](#) / [Lucrare de curs 6: Setul de instructiuni](#)

Started on Monday, 2 November 2020, 12:42 PM

State Finished

Completed on Monday, 2 November 2020, 12:47 PM

Time taken 5 mins 1 sec

Marks 1.00/5.00

Grade **2.00** out of 10.00 (20%)

Question **1**

Incorrect

Mark 0.00 out of 1.00

În ce registru este stocat restul împărțirii pentru un împărțitor pe 16 biți (de tip word)?

- a. AH
- b. AX
- c. DX
- d. EBX

✗

The correct answer is: DX

Question **2**

Incorrect

Mark 0.00 out of 1.00

Ce instrucțiune nu afectează niciun indicator de stare?

- a. push
- b. cmp
- c. dec
- d. shl

✗

The correct answer is: push

Question **3**

Correct

Mark 1.00 out of 1.00

În urma operației "rol AL, 3", rezultatul din registrul AL este 0x5B. Cât era valoarea inițială din AL?

- a. 0x6B
- b. 0xDA
- c. 0xB6
- d. 0xAD

✓

The correct answer is: 0x6B

Question 4

Incorrect

Mark 0.00 out of 1.00

În urma operației de shiftare, cât va fi rezultatul din registrul AL și ce valoare va avea CF (Carry Flag)?

mov AL, 0x76

shr AL, 5

- a. Valoarea din AL este 0xC0, iar CF = 1.
- b. Valoarea din AL este 0xC0, iar CF = 0. ✗
- c. Valoarea din AL este 0x03, iar CF = 0.
- d. Valoarea din AL este 0x03, iar CF = 1.

The correct answer is: Valoarea din AL este 0x03, iar CF = 1.

Question 5

Incorrect

Mark 0.00 out of 1.00

Ce flag se activează în urma instrucțiunilor de mai jos?

mov AL, 100

add AL, 45

- a. Zero Flag
- b. Carry Flag
- c. Overflow Flag
- d. Parity Flag ✗

The correct answer is: Overflow Flag

◀ Laborator 4

Jump to...

Lucrare de curs 7: Declararea datelor ►

Started on Friday, 6 November 2020, 8:32 PM

State Finished

Completed on Friday, 6 November 2020, 8:36 PM

Time taken 3 mins 16 secs

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question 1

Correct

Mark 1.00 out of 1.00

Care dintre următoarele instrucțiuni poate fi folosită pentru a implementa eficient funcția **strcpy** din C:

- a. cmpsb
- b. scasb
- c. movsb
- d. lodsb



The correct answer is: movsb

Question 2

Correct

Mark 1.00 out of 1.00

Adevărat sau fals: Folosirea instrucțiunilor pentru șiruri (**movs**, **scas** etc.) este **ÎNTOTDEAUNA** (pe orice procesor cu arhitectura x86, cuplat la orice periferice) mai rapidă decât folosirea instrucțiunilor generale (**mov**, **cmp** etc.).

- a. Fals
- b. Adevărat



The correct answer is: Fals

Question 3

Correct

Mark 1.00 out of 1.00

Care este diferența dintre instrucțiunile **sar** și **shr** în limbajul de asamblare x86?

- a. **sar** păstrează valoarea bitului celui mai **SEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 1
- b. **sar** păstrează valoarea bitului celui mai **NESEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 0
- c. **sar** păstrează valoarea bitului celui mai **SEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 0
- d. **sar** păstrează valoarea bitului celui mai **NESEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 1



The correct answer is: **sar** păstrează valoarea bitului celui mai **SEMNIFICATIV** al operandului destinație, pe când **shr** îl transformă în 0

Question 4

Correct

Mark 1.00 out of 1.00

Cum putem modifica direcția în care sunt parcurse de către instrucțiunea **movsb** șirurile ale căror adrese sunt stocate în **ESI** și **EDI**?

- a. Direcția de parcurgere a șirurilor este mereu "de la stânga la dreapta", adică în sensul creșterii adreselor din **ESI** și **EDI**
- b. Folosind instrucțiunile **std** sau **cld**
- c. Prefixand instrucțiunea **movsb** cu una dintre instrucțiunile **repe** sau **repne**
- d. Folosind instrucțiunea **cdq**



The correct answer is: Folosind instrucțiunile **std** sau **cld**

Question 5

Correct

Mark 1.00 out of 1.00

O linie de cod în limbaj de asamblare poate conține:

- a. Unul sau mai multe labeluri
- b. Cel mult o instrucțiune de jump
- c. Cel puțin o mnemonică
- d. Cel mult un operand



The correct answer is: Cel mult o instrucțiune de jump

[← Lucrare de curs 6: Setul de instrucțiuni](#)

Jump to...

[Laborator 5 ►](#)

Started on Sunday, 15 November 2020, 9:23 PM

State Finished

Completed on Sunday, 15 November 2020, 9:24 PM

Time taken 1 min

Marks 4.00/5.00

Grade **8.00** out of 10.00 (80%)

Question 1

Correct

Mark 1.00 out of 1.00

Care din următoarele reprezinta un exemplu de adresare BAZATĂ ȘI INDEXATĂ CU SCALĂ (bază + index * scala + deplasament)?

- a. mov eax, [edx + ebx*4 + 0x12345678]
- b. mov eax, [ebp + 8]
- c. mov eax, [0xabcddef00]
- d. mov eax, [esi + ebx + 0x8]



The correct answer is: mov eax, [edx + ebx*4 + 0x12345678]

Question 2

Correct

Mark 1.00 out of 1.00

Cum se declară corect o variabilă 'myVar' de tip BYTE neinițializată?

- a. myVar dw ?
- b. myVar db ?
- c. myVar db null
- d. myVar db undefined



The correct answer is: myVar db ?

Question 3

Correct

Mark 1.00 out of 1.00

Care din următoarele registre NU este registru general?

- a. SS
- b. ESI
- c. BH
- d. EBP



The correct answer is: SS

Question 4

Correct

Mark 1.00 out of 1.00

Alegeți specificatorul de tip invalid:

- a. HWORD (6 octeți)
- b. QWORD (8 octeți)
- c. DWORD (4 octeți)
- d. BYTE (1 octet)



The correct answer is: HWORD (6 octeți)

Question 5

Incorrect

Mark 0.00 out of 1.00

Care este instrucțiunea dintr-un macro care înmulțește cu 4 al doilea parametru al macro-ului?

- a. sal %2, 2
- b. sal %2, 4
- c. sal %1, 2
- d. mul %1, 4



The correct answer is: sal %2, 2

[◀ 2CC: Verificare: Moduri de adresare](#)

Jump to...

[Lucrare de curs 9: Moduri de adresare C2 + Stiva ►](#)

Started on Tuesday, 17 November 2020, 10:31 AM**State** Finished**Completed on** Tuesday, 17 November 2020, 10:33 AM**Time taken** 1 min 52 secs**Marks** 4.00/5.00**Grade** **8.00** out of 10.00 (80%)**Question 1**

Correct

Mark 1.00 out of 1.00

Cum se face transmiterea parametrilor unei funcții, în programele care folosesc sintaxa de apel cdecl pe o arhitectură x86 (32 de biți)?

- a. prin heap
- b. printr-un vector
- c. toate variantele sunt greșite
- d. prin stivă



The correct answer is: prin stivă

Question 2

Correct

Mark 1.00 out of 1.00

Să presupunem că avem o stivă goală și rulăm următoarea secvență de instrucțiuni: push 1; push 2; pop; push 3; pop; push 4; push 5; pop. Care va fi elementul din vârful stivei?

- a. 3
- b. 4
- c. 1
- d. 5



The correct answer is: 4

Question 3

Correct

Mark 1.00 out of 1.00

Care operație NU poate fi efectuată asupra stivei?

- a. pop
- b. index
- c. pushad
- d. push



The correct answer is: index

Question 4

Incorrect

Mark 0.00 out of 1.00

În general, care dintre variantele de mai jos este un dezavantaj al folosirii macrourilor?

- a. duc la o programare mai neficientă
- b. duc la o execuție mai lentă a programului
- c. se pot crea instrucțiuni noi
- d. fac codul mai greu de înțeles



The correct answer is: fac codul mai greu de înțeles

Question 5

Correct

Mark 1.00 out of 1.00

Cum se pot interschimba variabilele a și b folosind stiva?

- a. push b; push a; pop a; pop b
- b. push a; pop b; push b; pop a
- c. mov eax, a; mov a, b; mov b, eax
- d. push a; push b; pop a; pop b



The correct answer is: push a; push b; pop a; pop b

[◀ Lucrare de curs 8: Moduri de adresare](#)

Jump to...

[Curs 10-11 ►](#)

Started on Sunday, 22 November 2020, 11:57 AM

State Finished

Completed on Sunday, 22 November 2020, 12:01 PM

Time taken 4 mins 22 secs

Marks 1.00/5.00

Grade **2.00** out of 10.00 (20%)

Question **1**

Incorrect

Mark 0.00 out of 1.00

Ce va afișa funcția următoare, știind că a primit parametrii 1 și 2 în ordinea aceasta (push 1, push 2)?

myFunc:

```
pop eax
PRINTF32 `%d\x0`, eax
ret
```

- a. 2 ✗
- b. altă valoare
- c. nimic, pentru că va da eroare înainte de la instrucțiunea "pop"
- d. 1

The correct answer is: altă valoare

Question **2**

Incorrect

Mark 0.00 out of 1.00

Prin convenție, pe o arhitectură x86 (32 de biți), valoarea de return a unei funcții se plasează de regulă:

- a. în registrul esp
- b. în registrul eax (sau edx:eax, în cazul unui rezultat pe 8 octeți)
- c. pe stivă, urmând să fie preluată tot de acolo ✗
- d. într-o altă zonă de memorie (rodata, data, etc.)

The correct answer is: în registrul eax (sau edx:eax, în cazul unui rezultat pe 8 octeți)

Question 3

Incorrect

Mark 0.00 out of 1.00

Considerăm echivalentul în limbaj de asamblare al semnăturii următoare de funcție: "void myFunc(int arr[], int dim)". Dacă valoarea lui ebp este salvată pe stivă la începutul funcției, ce se găsește la adresa esp + 8?

- a. adresa de început a lui arr
- b. valoarea lui dim
- c. valoarea primului element din arr
- d. valoarea ultimului element din arr



The correct answer is: adresa de început a lui arr

Question 4

Incorrect

Mark 0.00 out of 1.00

Fie codul următor:

```
myFunc:
    push ebp
    mov ebp, esp
    sub esp, 2
    ...
```

La ce adresă se găsește valoarea primului parametru de pe stivă (pe o arhitectură x86)?

- a. esp + 10
- b. esp + 4
- c. esp + 6
- d. esp + 8



The correct answer is: esp + 10

Question 5

Correct

Mark 1.00 out of 1.00

Vrem să transmitem valoarea din registrul eax ca parametru unei funcții, fără a folosi instrucțiunea "push". O modalitate este:

- a. mov [esp], eax
sub esp, 4
- b. sub esp, 4
mov esp, eax
- c. mov esp, eax
sub esp, 4
- d. sub esp, 4
mov [esp], eax



The correct answer is: sub esp, 4
mov [esp], eax

[◀ Laborator 6](#)

Jump to...

[Laborator 7 ►](#)

Started on Friday, 27 November 2020, 5:28 PM

State Finished

Completed on Friday, 27 November 2020, 5:30 PM

Time taken 2 mins 8 secs

Marks 3.00/5.00

Grade **6.00** out of 10.00 (60%)

Question 1

Correct

Mark 1.00 out of 1.00

Fie următoarea semnătura de funcție "void func(struct point *p)" și structura:

```
struct point {
    int x;
    int y;
}
```

Cum se poate pune valoarea membrului y în registrul eax în interiorul funcției, presupunând că elementele structurii nu sunt modificate de compilator în privința ordinii și a spațiului dintre ele (padding)?

- a. mov eax, [ebp + 12]
 mov eax, [eax]
- b. mov eax, [ebp + 4]
 mov eax, [eax + 4]
- c. mov eax, [ebp + 8]
 mov eax, [eax + 4] ✓
- d. mov eax, [ebp + 8]
 mov eax, [eax + 1]

The correct answer is: mov eax, [ebp + 8]
mov eax, [eax + 4]

Question 2

Correct

Mark 1.00 out of 1.00

Ce date despre procedura curentă NU sunt puse pe stivă pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl?

- a. variabilele locale
- b. vechiul esp ✓
- c. parametrii
- d. adresa de return

The correct answer is: vechiul esp

Question 3

Incorrect

Mark 0.00 out of 1.00

Cu ce este echivalentă următoarea secvență în limbaj de asamblare?

```
push ebp  
mov ebp, esp  
sub esp, 4
```

- a. leave 4
- b. leave ✗
- c. enter 4, 0
- d. enter 1, 0

The correct answer is: enter 4, 0

Question 4

Incorrect

Mark 0.00 out of 1.00

Fie semnătura de funcție "void func(int a, int b, int c)". La ce offset față de ebp se află parametrii a, respectiv c pe un sistem pe 32 de biți?

- a. -4, -12 ✗
- b. +8, +16
- c. -8, -16
- d. +4, +12

The correct answer is: +8, +16

Question 5

Correct

Mark 1.00 out of 1.00

Fie următoarea funcție:

```
int func(int *arr, int size) {  
    short *n;  
    char *str;  
    int y;  
    ...  
}
```

Știind că variabila n se află la adresa X, la ce adresă se află variabila y pe o arhitectură x86 pe 32 de biți, pe un program care folosește sintaxa de apel cdecl? Considerăm că variabilele locale nu sunt reordonate de către compilator și că nu există niciun padding.

- a. X + 8
- b. X - 8 ✓
- c. X + 5
- d. X - 5

The correct answer is: X - 8

[◀ 2CC: Hammer Time](#)[Jump to...](#)[Lucrare de curs 12: C + asm ►](#)

Started on Wednesday, 2 December 2020, 5:57 PM**State** Finished**Completed on** Wednesday, 2 December 2020, 5:58 PM**Time taken** 50 secs**Marks** 4.00/5.00**Grade** **8.00** out of 10.00 (80%)**Question 1**

Correct

Mark 1.00 out of 1.00

Conform convenției de apel CDECL, o valoare de return pe 32 de biți se pune:

- a. în registrul ESP
- b. pe stivă
- c. în registrul EIP
- d. în registrul EAX



The correct answer is: în registrul EAX

Question 2

Correct

Mark 1.00 out of 1.00

Care dintre următoarele reprezintă un dezavantaj al folosirii limbajului de asamblare, comparat cu folosirea unui limbaj de nivel mai înalt?

- a. acces la operații low-level
- b. mentenanța
- c. memoria ocupată de cod
- d. slab control asupra memoriei programului



The correct answer is: mentenanța

Question 3

Correct

Mark 1.00 out of 1.00

Marcați cu adevărat (A) sau fals (B) afirmațiile: În general, inline assembly aduce optimizări programului scris în limbaj high-level/Numărul variabilelor folosibile în inline assembly este unul limitat

- a. A/F
- b. F/A
- c. F/F
- d. A/A



The correct answer is: A/A

Question 4

Incorrect

Mark 0.00 out of 1.00

Cu ce directivă trebuie prefixată declararea unui funcții simple C (antetul funcției), dar definită într-un alt modul C sau într-un modul în limbaj de asamblare?

- a. nu este nevoie de o directivă
- b. include
- c. global
- d. extern



The correct answer is: nu este nevoie de o directivă

Question 5

Correct

Mark 1.00 out of 1.00

Care registru NU poate fi modificat în mod direct de către programator?

- a. EIP
- b. EDI
- c. EBP
- d. ESP



The correct answer is: EIP

[◀ Lucrare de curs 11: Funcții \(2\)](#)

Jump to...

[Curs 13 ►](#)

Started on Friday, 11 December 2020, 9:54 PM

State Finished

Completed on Friday, 11 December 2020, 9:55 PM

Time taken 1 min

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele reprezintă o unealtă de analiză statică?

- a. objdump
- b. valgrind
- c. ld
- d. ltrace



The correct answer is: objdump

Question **2**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele variante de răspuns este adevărată cu privire la directiva de asamblare global?

- a. Directiva global marchează o etichetă accesibilă și din alte module ale programului.
- b. Este obligatoriu să specificam tipul etichetei.
- c. Directiva global se folosește doar pentru variabile.
- d. Directiva global se folosește doar pentru nume de proceduri.



The correct answer is: Directiva global marchează o etichetă accesibilă și din alte module ale programului.

Question 3

Correct

Mark 1.00 out of 1.00

Care dintre următoarele variante de răspuns este adevărată cu privire la unealta xxd?

- a. xxd poate fi folosit doar pentru codificarea în hexazecimal.
- b. xxd poate fi folosit atât pentru codificarea cât și pentru decodificarea în / din hexazecimal. ✓
- c. xxd poate fi folosit doar pentru decodificarea în hexazecimal.
- d. xxd poate fi folosit pentru eliminarea simbolurilor dintr-un executabil.

The correct answer is: xxd poate fi folosit atât pentru codificarea cât și pentru decodificarea în / din hexazecimal.

Question 4

Correct

Mark 1.00 out of 1.00

În care dintre următoarele situații unealta gdb se pretează cel mai bine?

- a. Vrem să aflăm simbolurile dintr-un executabil și ce siruri de caractere sunt definite în cadrul programului.
- b. Vrem să aflăm unde apare un memory leak.
- c. Vrem să aflăm valoarea registrelor într-un anume punct din execuția programului. ✓
- d. Vrem să aflăm mai multe metadate despre program.

The correct answer is: Vrem să aflăm valoarea registrelor într-un anume punct din execuția programului.

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre următoarele instrucțiuni poate fi folosită pentru a șterge simbolurile de debug dintr-un fișier binar?

- a. nm
- b. strings
- c. bc
- d. strip ✓

The correct answer is: strip

[◀ Laborator 8](#)

Jump to...

[2CC: Recap 14: Gestiunea bufferelor ►](#)

[Dashboard](#) / My courses / [03-ACS-L-A2-S1-IOCLA-CTI](#) / [14 December - 20 December](#) / [Lucrare de curs 14: Buffer Overflow, Securitate](#)

Started on Sunday, 20 December 2020, 11:37 AM

State Finished

Completed on Sunday, 20 December 2020, 11:38 AM

Time taken 1 min 13 secs

Grade 8.00 out of 10.00 (80%)

Question 1

Correct

Mark 2.00 out of 2.00

NU reprezintă o metodă de abuzare a bufferelor:

- a. Memory disclosure
- b. OOB (out of bounds)
- c. Canari
- d. Buffer overflow



The correct answer is: Canari

Question 2

Correct

Mark 2.00 out of 2.00

Este un utilitar folosit în detectarea abuzării bufferelor:

- a. nasm
- b. IDA
- c. nvim
- d. vsodium



The correct answer is: IDA

Question 3

Correct

Mark 2.00 out of 2.00

Se dă următoarea declarație a unui buffer în limbaj de asamblare: sub esp, 256. Sistemul este pe 32 biți. Care dintre următoarele declarații C sunt echivalente declarației în limbaj de asamblare de mai sus?

- a. unsigned short[128], int buf[256], long long buf[8]
- b. double buf[32], int buf[64], unsigned short[128]
- c. int buf[32], char buf[256], double buf[16]
- d. int buf[64], unsigned short buf[128], char buf[128]



The correct answer is: double buf[32], int buf[64], unsigned short[128]

Question 4

Correct

Mark 2.00 out of 2.00

Când poate apărea o eroare de tipul OOB (out of bounds)?

- a. Când se face bounds checking
- b. Când nu s-a apelat ret în cadrul unei funcții
- c. Când programul nu afișează rezultatul așteptat
- d. Când se fac operații dincolo de limita bufferului



The correct answer is: Când se fac operații dincolo de limita bufferului

Question 5

Incorrect

Mark 0.00 out of 2.00

Fie următoarea funcție:`void func(void) { double trouble; int buf[10]; ... }` Câți octeți sunt între adresa de început a lui buf și adresa de return a lui f?

- a. Nu știm, depinde de compilator.
- b. 48
- c. 44
- d. 18



The correct answer is: Nu știm, depinde de compilator.

[◀ Laborator 9](#)

Jump to...

Curs 16-17 ►

Started on Monday, 11 January 2021, 6:31 PM

State Finished

Completed on Monday, 11 January 2021, 6:36 PM

Time taken 5 mins 1 sec

Marks 4.00/5.00

Grade **8.00** out of 10.00 (80%)

Question 1

Correct

Mark 1.00 out of 1.00

Un buffer începe la adresa ebp-92. Cât este offsetul necesar pentru a suprascrie adresa de return, pe un sistem pe 32 de biți?

- a. 88
- b. 100
- c. 104
- d. 96



The correct answer is: 96

Question 2

Incorrect

Mark 0.00 out of 1.00

În următorul payload: offset * "Z" + "\x78\x56\x34\x12" + 4 * "Y" + "\xAB\xCD\xEF\xGH" + "\xBE\xBA\xFE\xCA", buffer-ul găsindu-se la adresa ebp-offset. Ultima valoare reprezintă:

- a. valoarea primului parametru al funcției
- b. valoarea celui de-al doilea parametru al funcției
- c. adresa funcției
- d. adresa de return a funcției



The correct answer is: valoarea celui de-al doilea parametru al funcției

Question 3

Correct

Mark 1.00 out of 1.00

Care dintre următoarele utilitare este folosit pentru analiză dinamică?

- a. GDB
- b. IDA
- c. objdump
- d. nm



The correct answer is: GDB

Question 4

Correct

Mark 1.00 out of 1.00

Fie următorul cod: char buffer[100]; fgets(buffer, len, stdin); Ce proprietate trebuie să aibă len pentru a NU se realiza buffer overflow?

- a. len > 100
- b. len ≤ 100
- c. len = 100
- d. Nu există o condiție pentru valoarea lui len



The correct answer is: len ≤ 100

Question 5

Correct

Mark 1.00 out of 1.00

Care dintre următoarele poate fi considerat un DEZAVANTAJ al analizei dinamice?

- a. nu poate fi realizată pentru orice aplicație (inflexibilitate)
- b. nu poate detecta vulnerabilități apărute la runtime (de exemplu segmentation fault)
- c. nu se poate realiza analiză dacă n-am acces la codul sursă
- d. în general, acoperirea (coverage) este mai redusă decât în cazul analizei statice



The correct answer is: în general, acoperirea (coverage) este mai redusă decât în cazul analizei statice

[◀ Laborator 10](#)

Jump to...

[Lucrare de curs 16: Optimizări ►](#)

Started on Saturday, 16 January 2021, 9:37 AM

State Finished

Completed on Saturday, 16 January 2021, 9:39 AM

Time taken 1 min 58 secs

Marks 5.00/5.00

Grade **10.00** out of 10.00 (**100%**)

Question **1**

Correct

Mark 1.00 out of 1.00

Care afirmație este FALSĂ cu privire la instrucțiunile de tip jmp/branch?

- a. predicțiile făcute de CPU în cazul salturilor de tip branch sunt întotdeauna adevărate ✓
- b. dacă avem doar o instrucțiune per loop, execuția datelor în pipeline nu mai are sens
- c. obținem îmbunătățiri dacă reducem din numărul instrucțiunilor de tip branch
- d. salturile la distanțe mari în cod fac imposibilă stocarea lui în cache

The correct answer is: predicțiile făcute de CPU în cazul salturilor de tip branch sunt întotdeauna adevărate

Question **2**

Correct

Mark 1.00 out of 1.00

Care dintre următoarele transformări NU reprezintă o optimizare?

- a. pentru împărțirea mai multor numere la x, calculez $1/x$ și după multiplic
- b. folosirea insert sort, în locul lui qsort din libc ✓
- c. folosirea instrucțiunilor de tipul CMOV în locul instrucțiunilor de tip branch
- d. folosirea shiftarilor în locul înmulțirii/împărțirii cu puteri ale lui 2

The correct answer is: folosirea insert sort, în locul lui qsort din libc

Question 3

Correct

Mark 1.00 out of 1.00

Care exemplu NU ilustrează aplicarea tehnicii loop unrolling?

- a. for ($i = 0; i < 100; i++$)

```
for (j = 0; j < 100; j++)  
    a[i][j] = b[i][j];
```

- b. for ($i = 0; i < 100; i++$)

```
for (j = 0; j < 100; j+=2) {  
    a[i][j] = b[i][j];  
    a[i][j+1] = b[i][j+1];  
}
```

- c. for ($i = 0; i < 100; i+=2$)

```
for (j = 0; j < 100; j++) {  
    a[i][j] = b[i][j];  
    a[i+1][j] = b[i+1][j];  
}
```

- d. for ($i = 0; i < 100; i+=2$)

```
for (j = 0; j < 100; j+=2) {  
    a[i][j] = b[i][j];  
    a[i][j+1] = b[i][j+1];  
    a[i+1][j] = b[i+1][j];  
    a[i+1][j+1] = b[i+1][j+1];  
}
```



The correct answer is: for ($i = 0; i < 100; i++$)

for ($j = 0; j < 100; j++$)

$a[i][j] = b[i][j];$

Question 4

Correct

Mark 1.00 out of 1.00

Cum pot fi optimizate operațiile matematice:

1. $x / 16$

2. x^4

3. x^3

- a. $x << 4 \quad | \quad x2 = \text{pow}(x, 2), x4 = \text{pow}(x2, 2) \quad | \quad x*x*x$

- b. $x >> 4 \quad | \quad \text{pow}(x, 4) \quad | \quad \text{pow}(x, 3)$

- c. $x >> 4 \quad | \quad x2 = x*x, x4 = x2*x2 \quad | \quad x*x*x$

- d. $x / \text{pow}(2, 4) \quad | \quad \text{pow}(x, 4) \quad | \quad x*x*x$



The correct answer is: $x >> 4 \quad | \quad x2 = x*x, x4 = x2*x2 \quad | \quad x*x*x$

Question 5

Correct

Mark 1.00 out of 1.00

Care afirmație este FALSĂ privind înmulțirea matricelor:

- a. deoarece capacitatea cache-ului este mare, este eficient dacă citim/încarcăm întreaga linie/coloană din matrice ✓
- b. algoritmul standard folosește 3 cicluri for imbricate, complexitatea fiind $O(n^3)$
- c. poate fi optimizat, folosind blocuri mai mici
- d. soluția optimizată folosește eficient memoria cache, mărindu-se altfel performanța programului

The correct answer is: deoarece capacitatea cache-ului este mare, este eficient dacă citim/încarcăm întreaga linie/coloană din matrice

[◀ Lucrare de curs 15: Buffer Overflow. Securitate \(2\)](#)

Jump to...

[Curs 18-19 ►](#)