

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 main() { 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

În 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 bibliotecă 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. Să 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 **probabil**
- ☐ 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](#)