SO 2020-2021: Întrebări examen oral

Aceste întrebări vor fi folosite pentru discuția de la examenul oral de SO. Nu sunt singurele întrebări care vor fi prezente în discuție; de la aceste întrebări discuția va atinge și alte noțiuni prezentate la cursul de SO.

Stiva software

Ce este un apel de sistem?
De ce sunt necesare apeluri de sistem?
Ce avantaj / dezavantaje au apelurile de sistem?
Ce înseamnă user/application mode/space (mod neprivilegiat)? Ce înseamnă kernel/supervisor mode/space (mod privilegiat)?
Cum se realizează tranziția în mod privilegiat?
Ce se întâmplă în momentul tranziției în mod privilegiat? Cum se/Cine asigură (enforcement) existența modului privilegiat?
Ce este o bibliotecă?
Care este asocierea apel de bibliotecă / apel de sistem?
Care este rolul bibliotecii standard C (libc)?
Ce acțiuni se pot executa doar în mod privilegiat?
Ce operații / instrucțiuni low-level (ISA) se pot executa doar în mod privilegiat?
Ce este un sistem de operare monolitic?
Ce este un sistem de operare de tip microkernel?
Care sunt avantajele unui sistem de operare monolitic?
Care sunt avantajele unui sistem de operare de tip microkernel?
Care tip de sistem de operare are mai multe apeluri de sistem?
Care este avantajul folosirii mașinilor virtuale din perspectiva securității?
Ce este o bibliotecă statică? Ce este o bibliotecă dinamică?
Când preferăm folosirea static linking, respectiv dynamic linking? Cu ce diferă un executabil dinamic de un executabil static?
Dați exemplu de apel de sistem blocant.
Dați exemplu de apel de sistem neblocant.
De ce, în general, o aplicație trebuie să execute un apel de sistem pentru a accesa un dispozitiv hardware? De ce NU poate accesa direct dispozitivul hardware?
Ce înțelegem prin overhead spațial și overhead temporal?
Dați exemplu de mecanism / funcție care reduce overhead-ul spațial și unul care reduce overhead-ul temporal.
Ce înseamnă double buffering? În ce situație concretă (mecanism/funcție) apare?
Ce se întâmplă când există o eroare critică (de tip Segmentation fault) la nivelul sistemului de operare?

Scheduling

Ce este un proces?
Ce este un thread?
Cu ce diferă un thread de un proces?
Cum este afectat spațiul virtual de adrese al unui proces în momentul creării unui thread?
Ce zone de memorie au comune thread-urile unui proces și ce zone au specifice?
Ce conține PCB (Process Control Block)?
Care sunt stările în care se poate găsi un proces/thread?
Ce efect are apelul fork()?
Ce resurse partajează/nu partajează procesul părinte și procesul copil în cazul apelului fork()?
Ce efect are apelul exec()?
Câte procese se pot găsi în starea RUNNING, READY și WAITING?
Ce este o schimbare de context? Ce se întâmplă la o schimbare de context?
Ce cauzează schimbări de context?
Ce este o schimbare de context voluntară și o schimbare de context nevoluntară?
Ce sunt thread-urile cu implementare user-level și thread-urile cu implementare kernel-level?
În ce situație este utilă zona TLS (thread local storage)?
De ce este necesară sincronizarea proceselor/thread-urilor?
Ce înseamnă race condition?
Ce înseamnă deadlock?
Ce înseamnă livelock? Cum diferă de un deadlock?
Care sunt dezavantajele sincronizării?
Ce înseamnă TOCTTOU (time of check to time of use)?
Când se blochează un producător în problema producător-consumator? Dar un consumator?
Cum se implementează pe un sistem single-core un spinlock? Cum se implementează pe un sistem multi-core un spinlock?
De ce este necesară prezența unei instrucțiuni de tipul atomic_compare_and_swap în fiecare ISA?
Cu ce diferă un spinlock de un mutex? Când folosim spinlock-uri? Când folosim mutex-uri?
Ce efect are folosirea operatorului & din shell în crearea unui proces?
Ce este un proces zombie? Cum apare un proces zombie? Care este problema proceselor zombie?
Ce este un proces orfan? De ce un proces este orfan foarte puțin timp?
Poate fi un proces zombie orfan? Ce se întâmplă cu un proces zombie orfan?
Ce se întâmplă dacă un thread realizează un acces invalid la o zonă de memorie?
Poate un thread să acceseze stiva altui thread? Cum?
De ce schimbarea de context între două thread-uri ale aceluiași proces este mai rapidă decât schimbarea de context între două thread-uri din procese diferite?
Ce forme de comunicare inter-proces cunoști?
Ce este un semnal? Când se trimite un semnal către un proces?

Cine trimite un semnal unui proces?
Cum este implementat operatorul | din shell?
Care sunt avantajele și dezavantajele folosirii memoriei partajate pentru comunicarea inter-proces?
Care sunt avantajele și dezavantajele pipe-urilor pentru comunicarea inter-proces?
Ce se întâmplă când toate procesele sistemului sunt blocate?
Ce înseamnă waiting time (timp de așteptare) în planificarea proceselor?
Putem avea un sistem multi-core cu un singur proces aflat în starea RUNNING și mai multe procese în READY?
Ce înseamnă starea READY/RUNNING/TERMINATED/WAITING(BLOCKED)?
Ce este un proces I/O intensive?
Ce este un proces CPU intensive?
Cum tratează planificatorul procesele I/O intensive și procesele CPU intensive?
Două thread-uri ale unui proces execută aceeași funcție. Care sunt diferențele între cele două thread-uri?
Ce este un apel thread-safe? Ce este un apel reentrant?
Cum tratăm situația în care apelăm o funcție non-reentrantă într-un handler de semnal?
Cu ce diferă procesul copil (creat prin fork) de procesul părinte?
Cu ce diferă un proces zombie de un proces orfan?
Ce parametri ai planificatorului trebuie să modificăm pentru a avea un sistem cu productivitate mai mare?
Ce parametri ai planificatorului trebuie să modificăm pentru a avea un sistem cât mai interactiv(responsive)?
Am avea nevoie de folosirea unui apel de sistem pentru crearea unui thread în cazul unei implementări de tip user-level threads? Dar în cazul deschiderii unui fișier în același scenariu?
Ce se întâmplă dacă folosim apeluri de sistem blocante în interiorul unui spinlock?
De ce este necesară folosirea prefixului LOCK pentru realizarea operațiilor atomice?

Memorie

Cum asigură sistemul de operare separația între procese?
Ce înseamnă mecanismul de memorie virtuală?
Ce reprezintă spațiul virtual de adrese al unui proces?
Ce este paginarea memoriei?
Ce este fragmentarea internă a memoriei?
Ce este fragmentarea externă a memoriei?
Ce rol are tabela de pagini?
Ce este și ce rol are MMU (Memory Management Unit)?
Ce rol are TLB?
Care este ordinul de mărime al numărului de intrări ale TLB?
Ce conține o intrare în tabela de pagini?
Ce înseamnă tabelă de pagini multi-nivel (ierarhică)? De ce este utilă?
Când are loc un TLB miss?
De ce se golește TLB-ul (TLB flush) la schimbare de context?

De ce nu este nevoie de TLB flush la schimbarea de context între două thread-uri ale aceluiași proces?
Ce înseamnă mecanismul de copy-on-write?
Dați exemple de situații în care are loc mecanismul de copy-on-write.
Cu ce apel de sistem asociem copy-on-write?
Când se duplică o pagină marcată copy-on-write?
Cine detectează un acces de scriere într-o pagină marcată copy-on-write?
Ce înseamnă demand paging?
În ce situație apare page fault fără a cauza segmentation fault?
Ce rol are spațiul de swap?
Când are loc swap in și swap out?
Care este rolul unui page fault. În ce condiții apare?
Care sunt secțiunile/zonele din spațiul de adrese al unui proces?
Ce secțiuni ale unui executabil se pot inspecta doar în timpul rulării?
Care sunt zonele writable din spațiul de adrese al unui proces?
De ce sunt avantajoase bibliotecile dinamice pentru spațiul de adrese al unui proces?
Două procese sunt pornite din același executabil, ce zone din spațiul de adrese vor partaja?
Se alocă un buffer a[100]. De ce a[105] NU va rezulta, în general, în Segmentation fault?
În ce situație a[300] rezultă în Segmentation fault?
Câte pagini fizice alocă un apel mmap() care alocă 1MB? O pagină ocupă 4KB.
Câte pagini fizice alocă un apel calloc() care alocă 1MB? O pagină ocupă 4KB.
Ce înseamnă maparea unui fișier în memorie? De ce este avantajos să mapăm fișiere față de folosirea read/write?
Câte page fault-uri se pot obține în cazul operației *a = b?
Care este numărul maxim de page fault-uri pe care îl poate genera expresia a = b + c?
Ce informații sunt reținute în stivă? Ce variabile C?
În ce zonă sunt reținute variabilele globale inițializate și cele neinițializate?
Ce înseamnă operația de stripping a unui executabil?
Ce se întâmplă la faza de loading (încărcarea unui executabil în memorie și crearea unui proces)?
Ce este entry point-ul într-un executabil?
Ce utilitare cunoașteți pentru analiză dinamică și ce utilitare cunoașteți pentru analiză statică?
Ce înseamnă analiză statică și ce înseamnă analiză dinamică?
Dați exemple de analizoare statice și analizoare dinamice.
Ce înseamnă Stack Guard / Stack Smashing Protection (SSP)?
Cu ce mecanism de protecție asociem funcția __stack_chk_fail?
Ce efect are ASLR (Address Space Layout Randomization)?
Ce efect are PIE (Position Independent Executable)?
Ce efect are PIC (Position Independent Code)?
La ce se referă un atac de tipul return-to-libc?
Ce înseamnă deturnarea fluxului de execuție a unui program (control flow hijack)? De ce este acest lucru relevant pentru un atacator?
Ce înseamnă memory leak / memory disclosure? De ce este acest lucru relevant pentru un atacator?
Ce înseamnă că o secvență de cod este PIC (Position Independent Code)?

De ce în general, preferăm o împărțire a spațiului virtual de adrese între kernel space și user space? Și nu un spațiu dedicat pentru kernel space?
Ce este un code pointer? De ce este interesant din perspectiva securității memoriei?
Ce este un shellcode?
Ce înseamnă code reuse din perspectiva securității memoriei?
Ce înseamnă shell injection din perspectiva securității memoriei?
Ce secvență de cod C va duce la o excepție de acces la memorie (de tip Segmentation fault)? De ce?
Cu ce diferă o funcție de o variabilă într-un executabil și/sau în cadrul spațiului de adrese al unui proces?
Două procese partajează o zonă de memorie. Cum se manifestă acest lucru în tabelele de de pagini ale celor două procese?
Putem avea mai multă memorie fizică decât dimensiunea maximă a spațiului virtual de adrese al unui proces? Dar invers?
Ce zone de memorie se aloca static? Dar dinamic?
Ce se întâmplă cu o variabilă modificată într-un proces copil din perspectiva procesului părinte?
Ce reprezintă un loader? Ce rol are acesta?
La ce folosim apelul mprotect? La ce mecanism de securitate putem face bypass folosind acest apel?
În ce zonă de memorie se află o variabilă globală, inițializată cu valoarea 0?

Fișiere, I/O

Ce conține un FCB (File Control Block)?
Ce reprezintă un descriptor de fișier?
Ce reprezintă tabele de descriptori de fișiere?
Câte tabele de descriptori de fișiere se găsesc într-un sistem de operare?
Ce efect are apelul dup()?
Ce efect are apelul close()?
Ce apeluri modifică pointer-ul/cursorul de fișier (file pointer)?
Ce apeluri modifică dimensiunea fișierului?
Ce efect are apelul/comanda truncate?
Ce este un hard link?
Ce este un link simbolic/symlink?
Care este diferența dintre un link simbolic și un hard link?
De ce numele unui fișier nu se găsește în inode?
Ce se întâmplă în cazul formatării unei partiții?
Ce se întâmplă cu sistemul de fișiere în cazul folosirii cu succes a comenzii rm?
Care este un avantaj al folosirii hard link-urilor și un avantaj al folosirii link-urilor simbolice?
Ce efect are comanda mv /path/to/a.dat /new/path/to/b.dat în sistemul de fișiere?
Care sunt tipurile de fișiere pe un sistem de fișiere uzual Unix?
Care tipuri de fișiere nu au blocuri de date?
Ce conțin blocurile de date ale unui director?
Ce este un sistem de fișiere virtual?

Ce este un dispozitiv virtual?
Ce tipuri de dispozitive cunoașteți? Clasificați-le din orice punct de vedere cunoașteți
Cu ce diferă un dispozitiv de tip bloc de un dispozitiv de tip caracter? Dați câte un exemplu de fiecare.
De ce nu are sens operația de seek pe un dispozitiv de tip caracter?
Ce adresă IP locală și ce port local are un socket întors de apelul accept()?
Ce valoare poate întoarce un apel read() sau un apel write()?
Ce operații se pot face pe fișiere?
Ce operații asupra fișierelor modifică/nu modifică valoarea cursorului unui fișier?
Ce operații asupra fișierelor modifică/nu modifică dimensiunea fișierului?
Unde este reținută valoarea cursorului de fișiere (file pointer) și unde este reținută dimensiunea fișierului?
De ce avem două buffere asociate fiecărui socket, ce rol are fiecare?
Ce este o întrerupere? Când este livrată o întrerupere?
Cu ce diferă port-mapped I/O de memory-mapped I/O?
Ce este o operație asincronă?
Ce este o operație neblocantă?
Cu ce diferă un socket de rețea de un socket UNIX?
Care este diferența între un pipe anonim și un pipe cu nume (named pipe)?
Ce este buffer cache-ul? Care este rolul său?
De ce operația write pe fișiere este foarte rar blocantă?
În ce situație operația read() pe fișier se blochează?
Care este rolul unui device driver?
Ce rol are controller-ul hardware?
Care este rolul DMA-ului (Direct Memory Access)?
Când are sens să folosim polling în loc de întreruperi?
Ce înseamnă zero-copy? Ce mecanism/apel folosește zero-copy?
Ce rol are mecanismul de TCP offload engine?
Care este sursa primară pentru care un apel send() pe un socket TCP se blochează?
Care este sursa primară pentru care un apel send() pe un socket UDP se blochează?
Ce garanții ni se oferă în momentul în care apelul send() se întoarce în user space?
Cu ce diferă afișarea folosind printf() față de folosirea write()?
De ce subsistemul de networking nu folosește buffer cache-ul?
Ce rol are apelul / comanda sync?
Care este rolul apelului ioctl / DeviceIoControl?
De ce în general doar utilizatorul root are permisiuni de scriere (uneori doar root are permisiuni de citire) pe intrările din /dev?
Ce permisiuni are zona .text/.data/.rodata/.bss/de stivă/ de heap?
De ce este apelul fwrite mai rapid decât write atunci când facem multe scrieri?
Ce se întâmplă dacă facem open de mai multe ori consecutiv pe același fișier?
Ce fișiere sunt deschise, în general, la crearea unui proces nou?
De ce este utilă prezența unor dispozitive pur virtuale în ierarhia /dev (ex. /dev/vboxnetctl, /dev/urandom)?
Ce conține tabela vectorilor de întrerupere / interrupt descriptor table?
Ce utilitar putem folosi pentru crearea unui hard link al unui fișier? Dar al unui director?
De ce nu se păstrează numele fișierului în inode?

Ce informații conține un directory entry (dentry)?
De ce, în general, ln permite crearea de link-uri simbolice pentru un director, dar nu și crearea de hard link-uri?