

1.Памет

- Всички процеси използват памет, дори и спящите, които използват тяхната си локална памет.
- Физическите свойства на паметта са: енергонезависима, време за достъп и многократно използване.
- Локалните данни, с които работи процеса трябва да са си негови и да са предпазени.
- Процесора има два режима:
 - Real mode - програмата има достъп до цялата памет
 - Protected mode - програмата няма достъп до цялата памет, а само до тази, която и е предоставена, както и до някои инструкции
 - Проста техника(Сегментация): за паметта, която е представена на всеки процес имаме два регистъра : начало(f) и дължина(s) и проверяваме адреса на всеки байт, който използва този процес дали е в този интервал т.е $f \leq \text{address} < f + s$
 - IBM 360 техника(Paging): имаме n страници и всяка страница има атрибути, в които е записано как и кои процеси могат да използват тази страница
- В съвременната система:
 - Аналог на Paging, но с доста свойства и хардуер, който да го поддържа и е известно с името виртуална памет(VM)
 - Виртуалната памет е абстрактно понятие
 - Всеки процес си има VM
 - Няма съответствие между VM и RAM
 - На някои страници от VM е съпоставена страница от RAM, на други може нищо да не е съпоставено, а на трети може да е съпоставена страница от HDD
 - На всяка наша страница с данни се съпоставя страница от RAM и я използваме монополно
 - Може да има страници от RAM, които се използват от няколко процеса, но тези страници трябва да са само за четене. Затова трябва пак да има информация за всяка страница как ще се ползва т.е трябва да има тагове
 - Към всяка страница от VM има тагове, които казват къде е записана тя в RAM или HDD и казва за какво ще се ползва тази страница
 - За всяка страница от VM трябва да знаем къде се намира тя реално и тага и тази информация се съдържа в таблица, която се нарича пряка адресна таблица, което означава, че за всеки процес имаме масив, който описва всяка негова страница
 - Има и друг вариант да се поддържа обща структура, която да показва тези данни за страниците от VM на всеки процес и тогава ключът в тази таблица ще е : <номер на страницата, номер на процеса>, тази таблица се нарича обратна таблица

- Memory management unit - в съвременния процесор го има и се грижи за адресните таблици, които са масиви някъде в паметта и има регистър, който сочи към тях
- 3 вида кеш:
 - Кеш за инструкции - малко на брой дълги интервали, само се чете
 - Кеш за данни - четене и писане на данни
 - Кеш за таблиците за управление на виртуална памет
- TLB - Translation Lookahead Buffer
 - Кешове за няколко отделни таблици за управление на виртуална памет - съдържа 32-битови думи
 - Не е с голям размер
 - Има две нива
- При процесите делим паметта между тях и даваме на всеки, колкото му трябва, но те са анонимни, те са парчета информация
- При файлове парчетата от памет трябва да са именувани и централния въпрос е за имената, когато файла пази информация трябва да е в устойчивата памет. Когато променяме информация във файла, трябва той да е свързан с процес
- Паметта можем да я представим, като множество от файлове - дълготрайно живеещи в информационната среда, информационни обекти
- Тъй като файловете ще съществуват дълго(те са споделяни между различни потребители и процеси) са важни имената им

2. Реализация на файлова система

- Твърд диск - диск разделен на много пътечки
 - над повърхнината има електрическо устройство, което засича дали бита е 1 или 0
 - Придвижването на главата от една пътечка на друга е бавно
 - Всяка пътечка е разделена на няколко сектора от по 512 или 1024 байта
 - Главата не се допира до диска, а лети над повърхността
 - Софтуера знае колко е голям сектора и колко е времето от един до друг
 - Файла се представя като много сектори
 - Не са последователно разпределени байтовете на файловете, защото така не могат лесно да нарастват
 - Файлът се разполага там, където може, процесът на разхвърляне се нарича фрагментация
- Алгоритъм на асансьора
 - Разместват се така заявките към четене и писане, че главата да изминава минимално разстояние
 - Има две приоритетни опашки: едната се състои от файлове, които се намират по посока на движението на главата, а другата от файлове, които се намират в обратната посока
 - Ако е празна главата на опашката по посока на главта, то главата си сменя посоката

- Двете стратегии - кеширане и Алгоритъм на асансьора пораждат проблем за надеждността (внезапно спиране на тока)
- Журнал
 - Записваме промените във файл, който се нарича журнал, в него се записва пълната информация за операциите над файлове и когато се позапълни/препълни журнала, спираме и отразяваме операциите от журнала над файловете, ако спре тока журнала ще пази последните промени, а файловете ще са си консистентни. Четат се първо старите файлове и се проверява дали в журнала са променени, ако спре тока, докато пишем операцията/транзакцията, тя няма да се е изтрила преди да е завършила и затова ще я пазим все още в журнала.
 - Във файловите системи транзакцията е елементарна файлова операция (read, write, open)
 - Журналът(log файла) се намира или в друг диск или друг дисков дял, за да може двете паралелно да работят, но в персоналните устройства журналът е файл в самата файлова система или в същия дял
 - Две фази
 - Файловите операции в истинския ред по време се записват в журнала
 - Когато се натрупат се подават заявките на алгоритъма на асансьора
 - Работата на журнала позволява да запазваме цялостта на диска и да се предпазим от разрушаване на структурите на диска.