



Жадыны басқару

- Орындаған: Сәденов Жандос

Жады және жадыны басқару

- Жады- мультибағдарламалы операциялық жүйе арқылы реттеліп, басқаруды талап ететін ең маңызды ресурстардың бірі болып табылады. Жадының бос бөліктерінің барлығы бөлістіруді қажет етеді. ОЖ жадыны басқаруда мынадай қызметтерді атқарады:
 - бос және толған жады бөлігін анықтау;
 - жадыны үрдістерге бөлу мен олардың аяқталуы кезінде жадыны босату;
 - үрдістерді оперативті жадыдан қатты дискіге ығыстыру;
 - оперативті жадыда орын босаған кезде оларды қайтара көшіру;
 - бағдарлама мекен-жайларын жадының нақты облысына орналастыру



Жадыны басқарудың талаптары

- Қайта орналастыру
- Қорғау
- Бірге қолдану
- Логикалық ұйымдастыру
- Физикалық ұйымдастыру

Қайта орналастыру

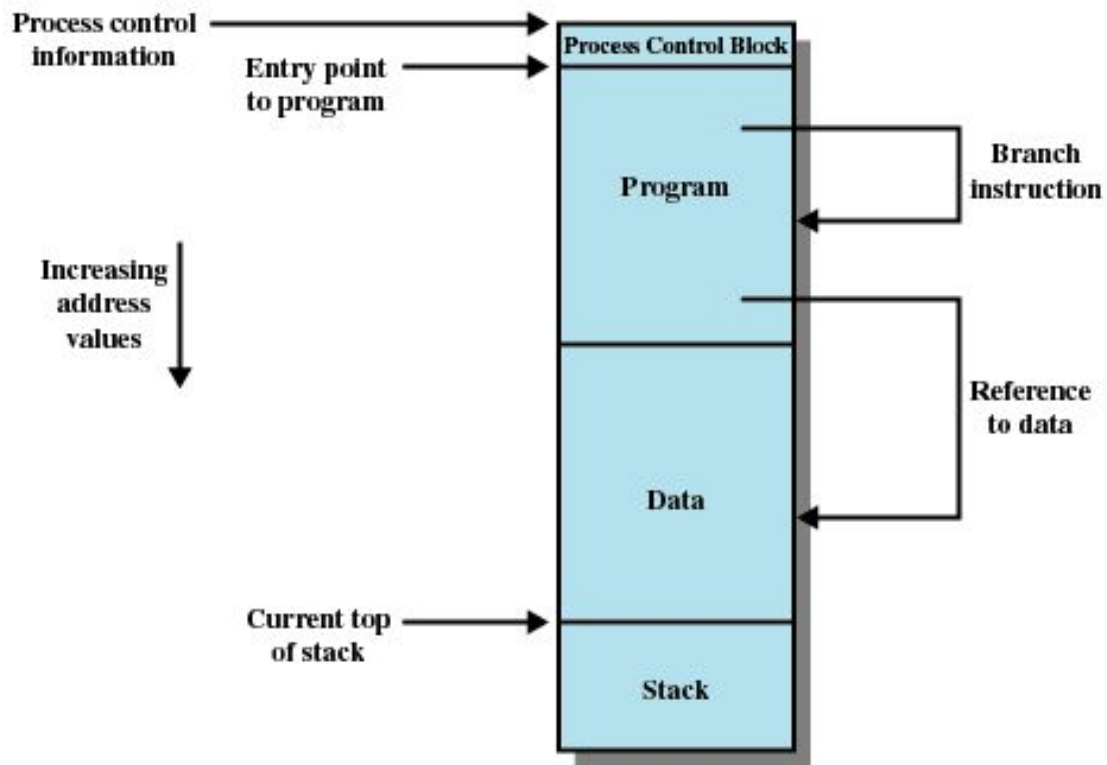


Figure 7.1 Addressing Requirements for a Process

Жадыны басқару

- Жадының иерархиялық құрылымы
- Жады менеджері (Жадыны басқару модулі)
- Жады менеджерінің мақсаты:
 - Дәл осы уақытта жадының қандай бөлігі қолданып тұрғанын, ал қандай бөлігі бос екенін бақылау
 - Қажетті жағдайда процестерге жадыдан орын бөледі және олардың аяқталуында ресурстарды босатады
 - Егер жады барлық процестерді сыйдыруға аз болса, онда ОЖ мен диск арасындағы мәліметтерді алмасуды басқарады

Жадыны басқару жүйесінің екі класы

- Орындалу кезіндегі ОЖ мен диск арасында алмасуды қолданатын процестер
 - Свопинг (swapping) механизмы
 - Виртуалды жады (virtual memory) механизмы
- ОЖ мен диск арасында алмасуды қолданбайтын процестер

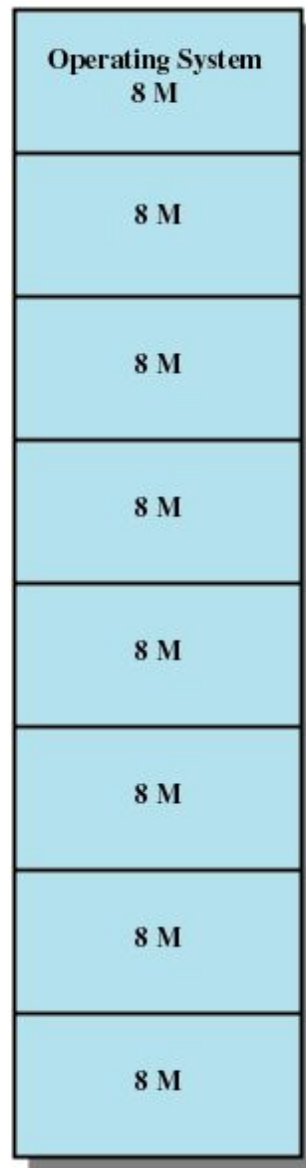
Дискке жүктелмейтін біресепті жүйе

- Уақыттың әрбір сәтінде тек қана бір бағдарлама орындалады, бұл кезде жады бағдарламалар арасында бөлінеді.

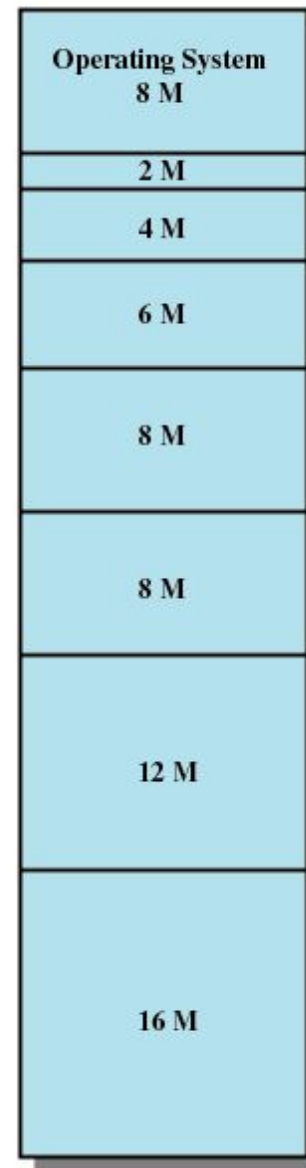


Бекітілген бөлімдермен берілген көпесептілік

- Бекітілген өлшем бойынша жүйе ОЖ-ны бірнеше бөлімдерге бөледі.
- Әр бөлімде бір есептен орналасып отырады.
- Жүйе есептер арасында процессорды жылдам ауыстырып қосады.



(a) Equal-size partitions



(b) Unequal-size partitions

Figure 7.2 Example of Fixed Partitioning of a 64-Mbyte Memory

Жадыны бекітілген түрде тарату. Орналастыру алгоритмі

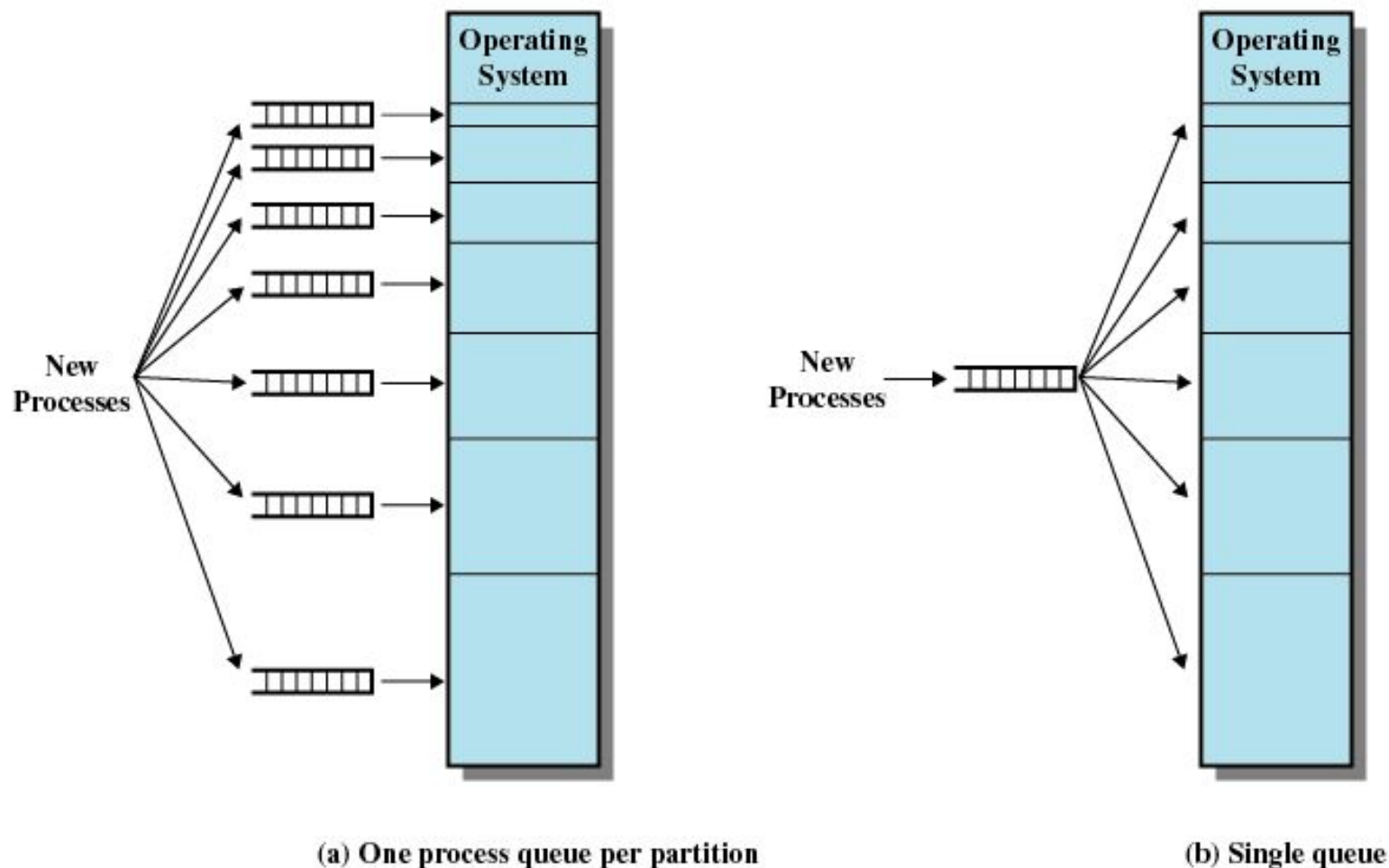


Figure 7.3 Memory Assignment for Fixed Partitioning

Динамикалық тарату

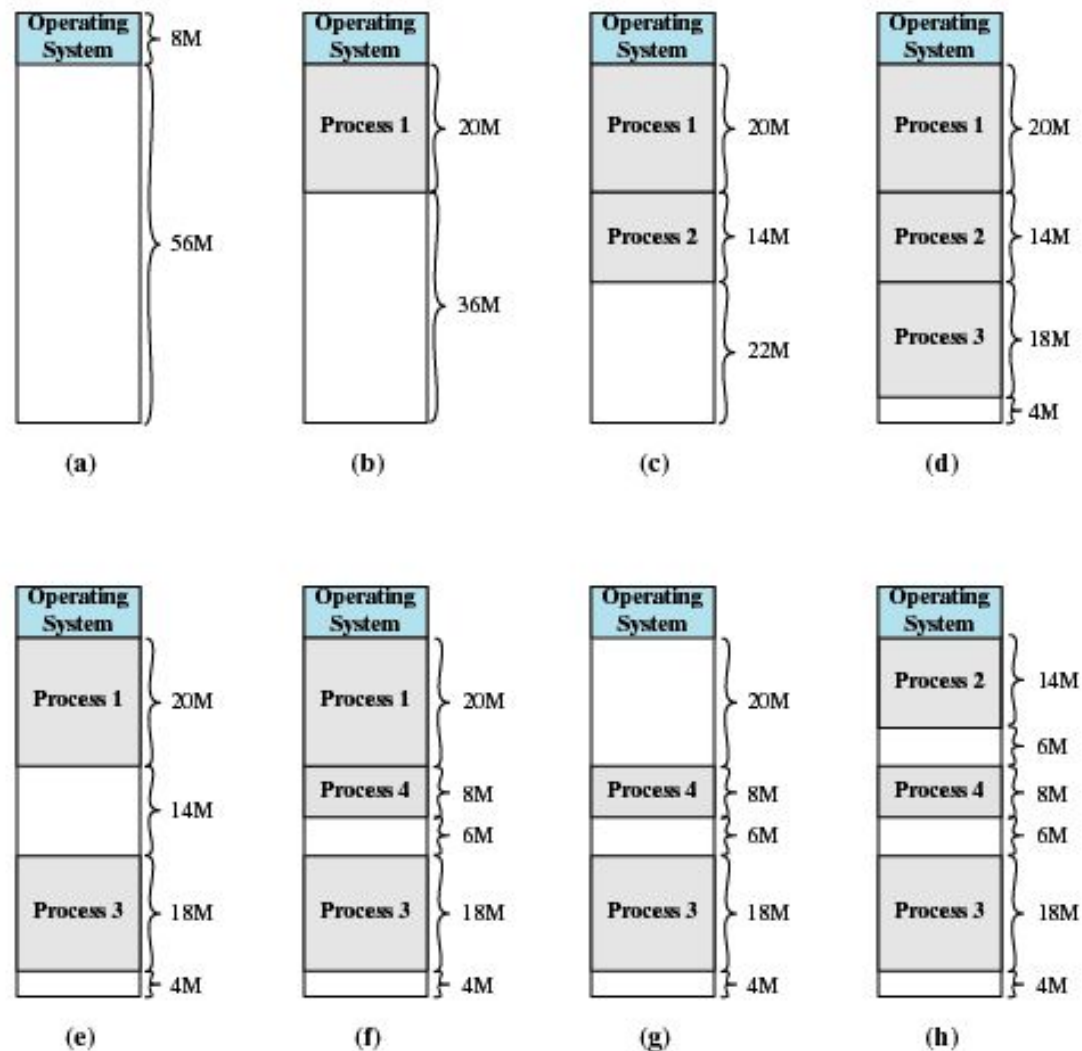
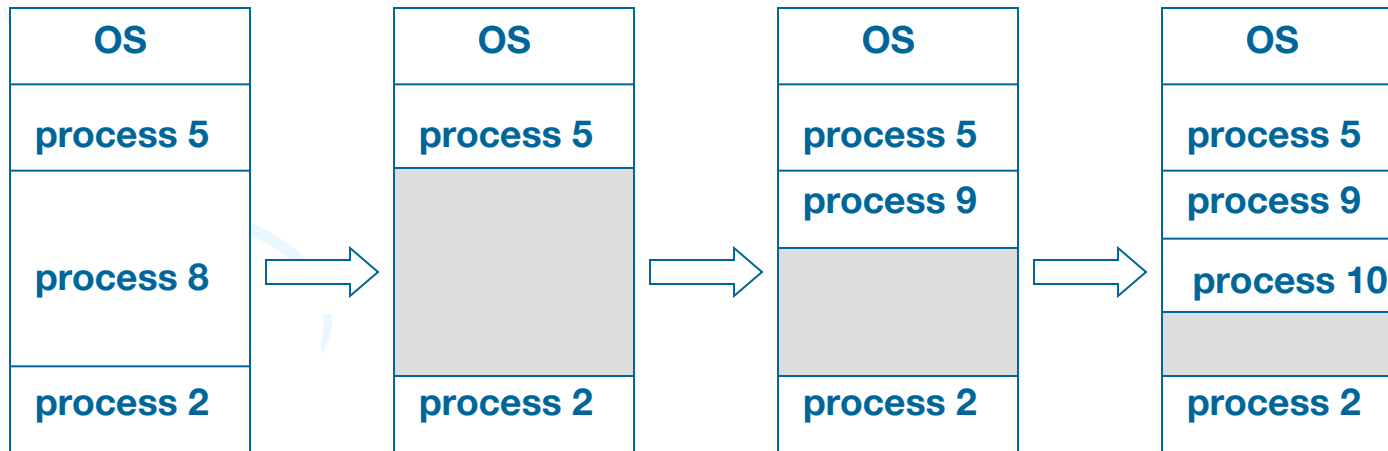


Figure 7.4 The Effect of Dynamic Partitioning

Динамикалық тарату

- Көптік тарату

- *Hole* (бос орын) –бұл жадыдағы әр түрлі көлемдегі орналасқан бос орындар, жадының бос аумағы.
- Процесс жүктелген кезде өзіне қолайлы бос аймақты алады.
- Динамикалық түрде процестердің келуі және аяқталуы кезінде саны, орналасуы және бөлімдерінің көлемі өзгереді.



Процеске жадыны берудің алгоритмдері

Егер процестер адрес бойынша сортталған және тізімде сақталған болса, онда алгоритмдер қатары бар болады.

- **Ең жақсы лайықты әдіс:** көлемі талапқа сай келетін бос орынды таңдау.
- **Бірінші лайықты әдіс :** Жадыны басынан бастап барлық бос орындарды тексереді және көлемі бойынша сай келетін алғашқы бос орынды таңдайды.
- **Келесі лайықты әдіс:** блоктың соңғы белгіленген жерінен бастап барлық бос орындарды тексереді және көлемі бойынша сай келетін алғашқы бос орынды таңдайды.

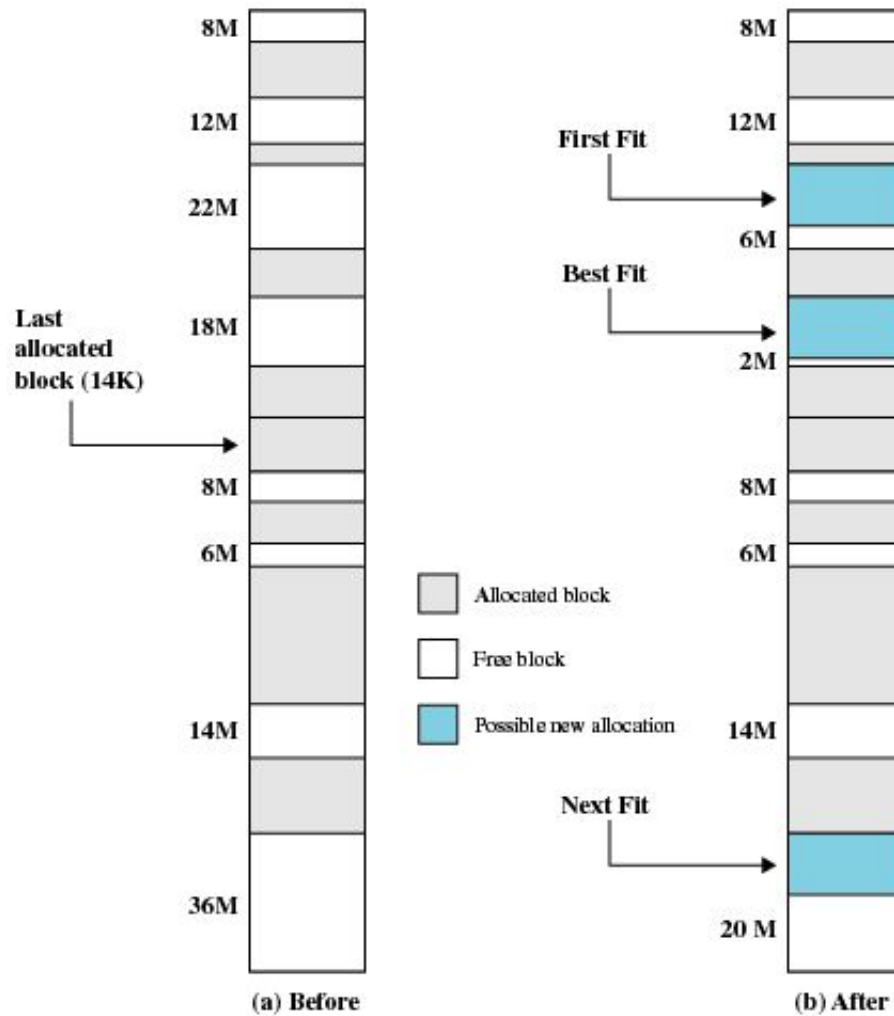


Figure 7.5 Example Memory Configuration Before and After Allocation of 16 Mbyte Block

Үзінділеу (Fragmentation)

ОЖ(оперативті жады)-ның қолдануға келмейтін кейбір бос аумақтары

- **Сыртқы үзінділеу (фрагментация)** – бос орындардың жалпы көлемі жаңа процесті орналастыруға жетлікті, бірақ процестерді орналастыруға бос орындардың бөліктері жеткіліксіз.
- **Ішкі үзінділеу (фрагментация)** - бұл процесс орындалады , егер процестің бағдарламалық бөлігінің көлемі және оның мәліметі болімнің көлемінен кіші болса.
- Сыртқы үзінділеудің кішірейуі
 - Жетілу– көршілес бос аумақтарды біріктіру арқылы үлкен көлемдегі бос орын құратын процесс
 - Жадыны нығыздау – процестер жадының адрестік кеңістіктің бас жағына орналасады

Адрестер

- **Логикалық адрес**

- Жадыдағы мәліметтердің ағымдағы орналасуына тәуелсіз жадының ұяшықтарына сілтемені көрсетеді
- Алдымен осы жадының ұяшығына қатынау алу үшін логикалық адресті физикалық адреске міндетті түрде трансляциялау керек

- **қатыстық адрес (Relative)**

- Логикалық адрестің жеке жағдайы
- Адрес кейбір белгілі нүктелердің тұрған жері бойынша анықталады(мысалға, программа басы)

- **Физикалық адрес (Physical)**

- абсолютті адрес немесе уақытша жадының нақты тұрған жері.

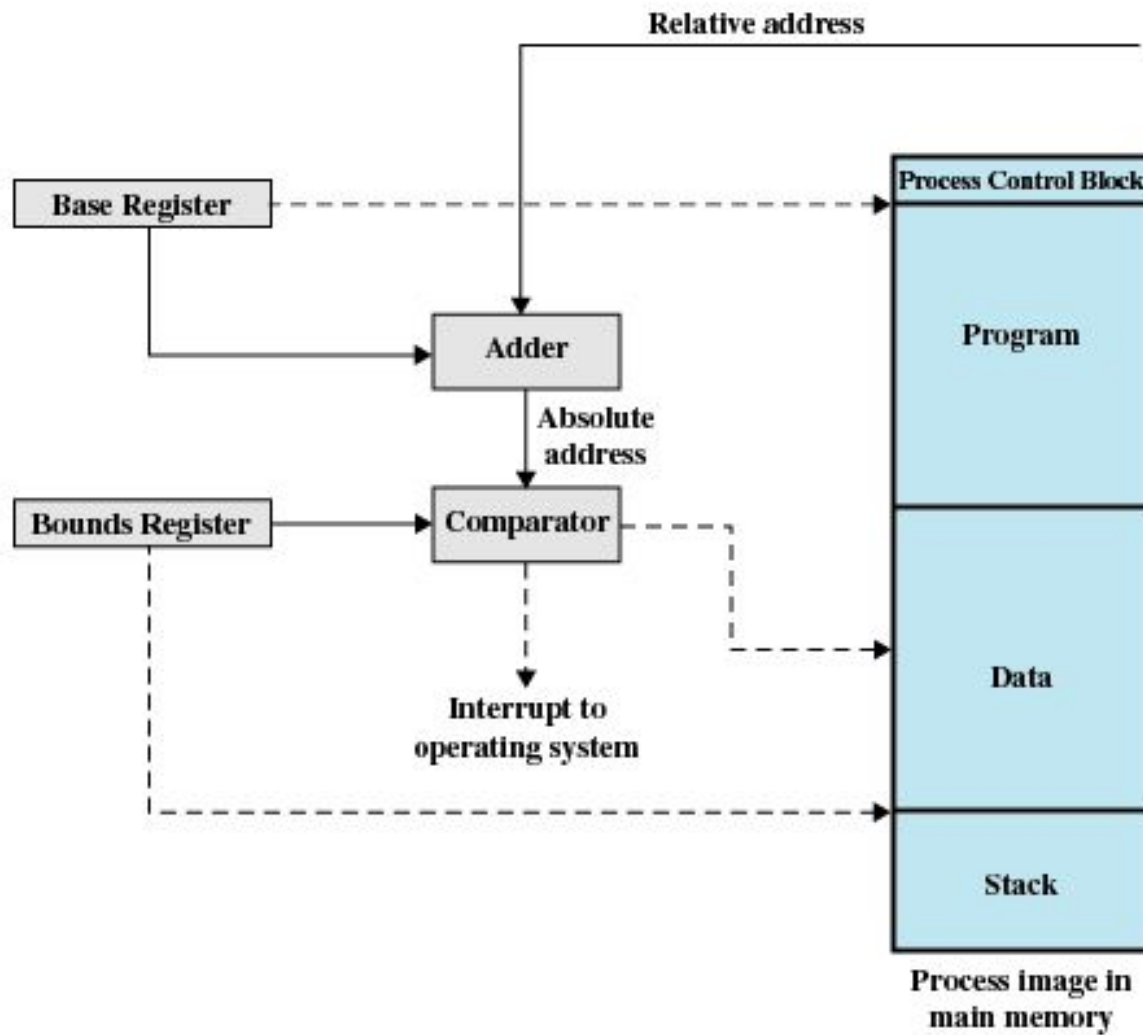


Figure 7.8 Hardware Support for Relocation



Арнайы аппараттық регистрлер

физикалық адресті салыстырмалы адреске трансляциялауға арналған аппараттық механизм

- Базалық регистрлер
 - Жадының басты бөлімде жүктелетін адрес
- Шектелген регистрлер
 - Бөлім ұзындығы



Арнайы аппараттық регистрлер

- Физикалық (абсолютный) адресті табу үшін базалық регистрдің мәні салыстырмалы адреске қосылады.

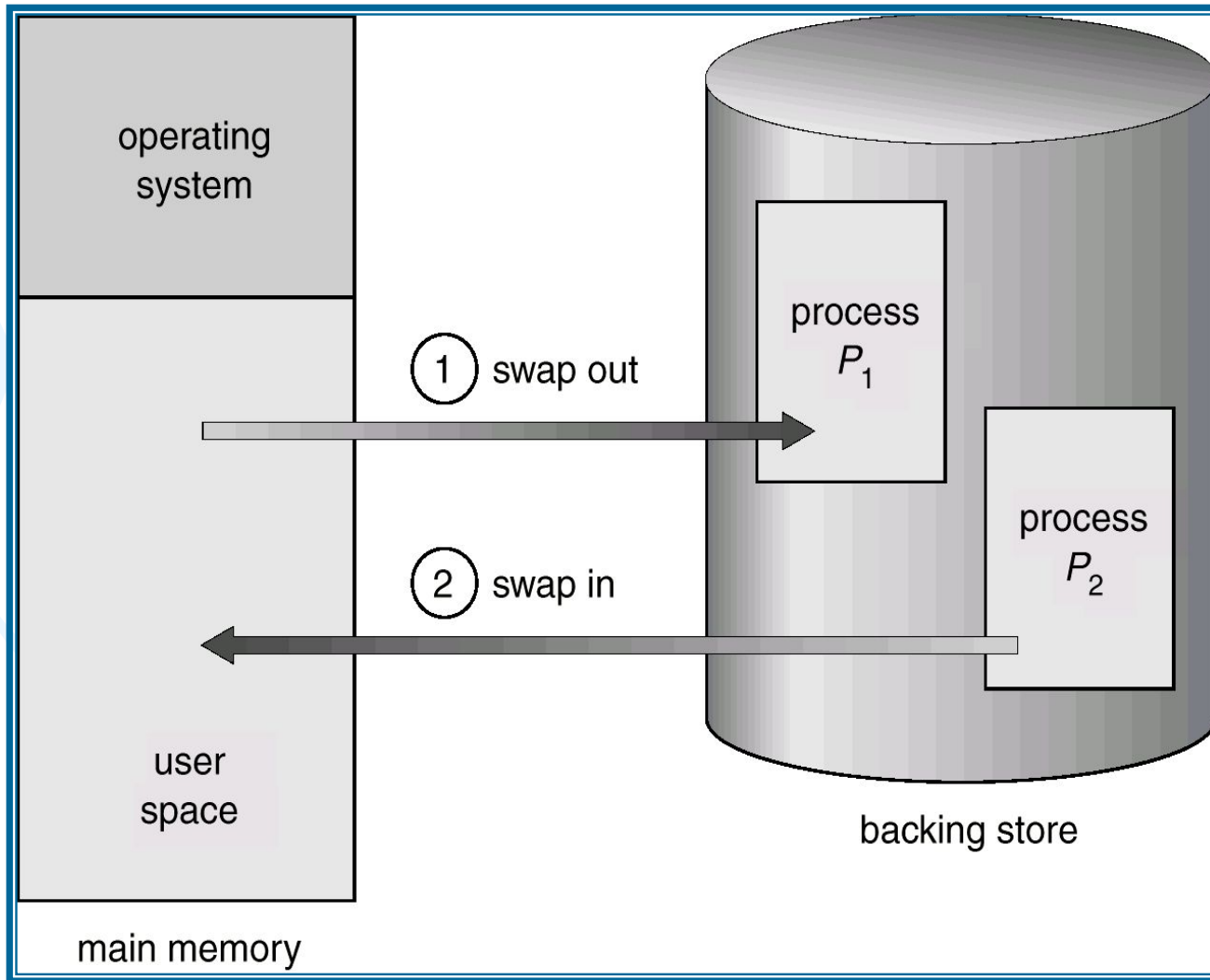
Жадыны басқарудағы негізгі жолдары

- **Свопинг** – әрбір процесс толығымен жадыға орналасады, біпнеше уақыт жұмыс жасаған соң қайтадан кері қайтарылады (жүйе УЖ дағы процестердің мәндерін жүктейді және оны жұмыстың орындалу барысында да бірнеше рет жүктейді).
- **Виртуалды жады** – программаға уақытша жадыда тек бөліктері ғана тұрса да орындалуға мүмкіндік бередіпозволяет.

СВОПИНГ

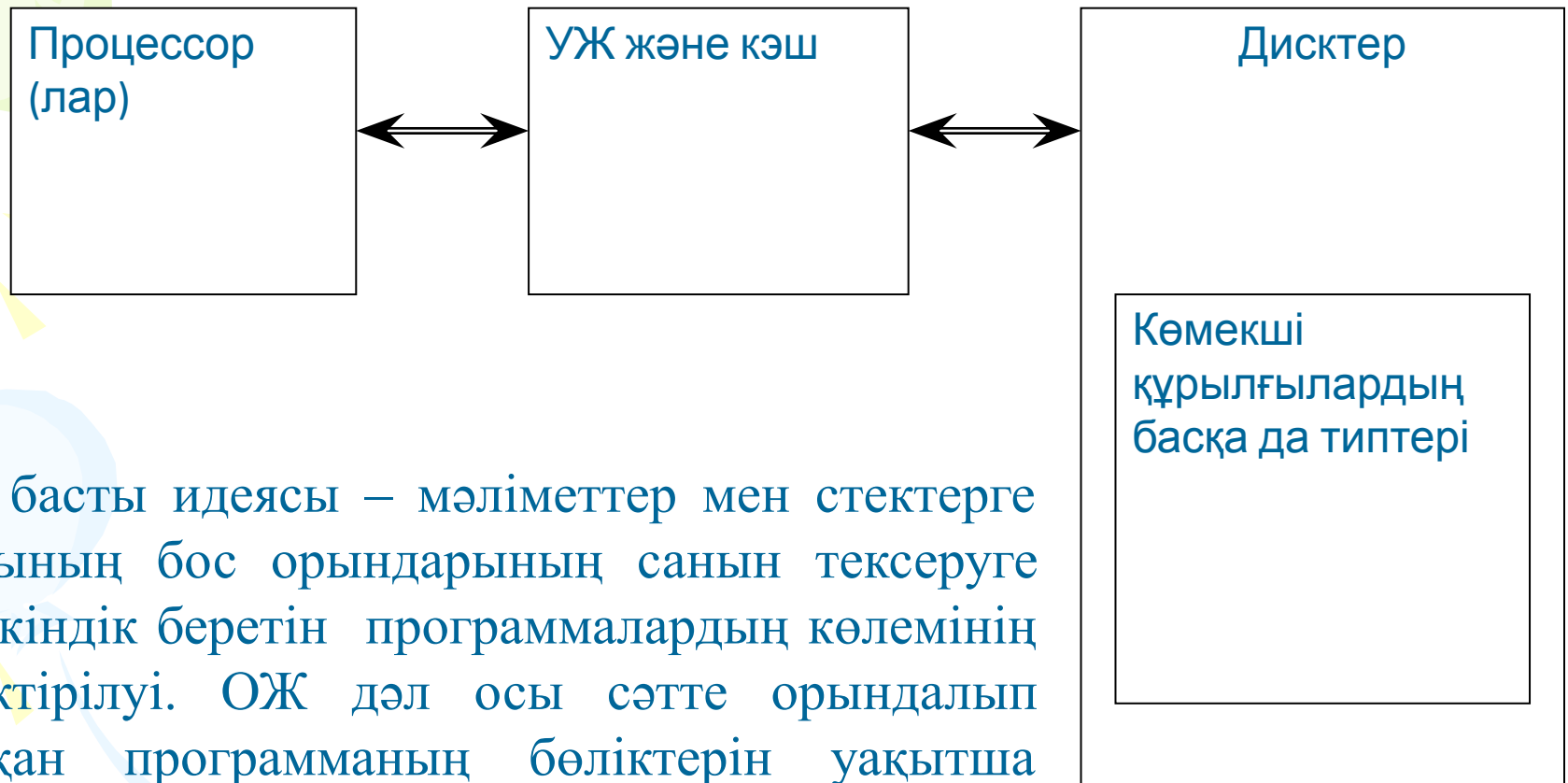
- Біруақытта уақытша жадыда бірнеше процестер бола алады.
- Процесс орындала береді, егер
 - Енгізу/шығаруға сұраныс генерацияланса
 - Уақыт интервалы аяқталғанша
 - Өздігінен аяқталады
- Сосын жүйе жады аймағындағы мәндерді екінші құрылғыға көшіре отырып, уақытша жадыдан процесті жүктейді.
- Жүйе уақытша жадына аймағындағы оның образын салыстыра отырып тербеліс аймағына уақытша жадыдағы келксі процесті жүктейді. Жаңа процесс жадыдан шығарылмағанша орындала береді, содан соң ғана келесі процесс жұмысын бастайды және т.б.

СВОПИНГ





Нақты	Нақты		Виртуалды		
Бірқолданушылық мамандандырылған жүйе (Однопользовательские специализированные системы)	Нақты жадыдағы мультипрограммалық жүйе		Виртуалды жадыдағы мультипрограммалық жүйе		
	Жадыны бекітілген таратудың жүйесі		беттік	сегменттік	Сегментті-беттік
	абсолютті	орналасатынын			



ВЖ басты идеясы – мәліметтер мен стектерге жадының бос орындарының санын тексеруге мүмкіндік беретін программалардың көлемінің біріктірілуі. ОЖ дәл осы сәтте орындалып жатқан программаның бөліктерін уақытша жадыда сақтайды, ал қалғаны дискіде сақталады. Жадыда және дискте орналасқан программа бөліктері қажет кезінде өзгертіледі.



Беттік ұйымдастыру

- **Беттік** - көлемдері бірдей болатын блоктарды **беттік** деп айтады
- **Сегменттік** – көлемдері әр түрлі болатын блоктар – **сегменттер**



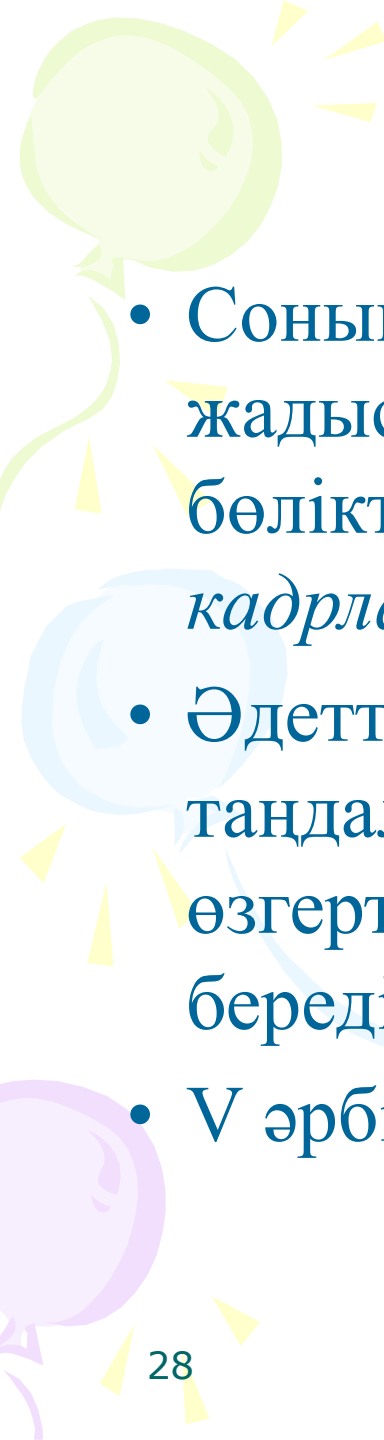
ВЖ ны ұйымдастыру

- V әрбір кадр = V әрбір бет = 2^k



Виртуалды беттер

- *Виртуалды бет* деп - виртуалды адрестік кеңістікте әрбір процесстердің берілген көлемдер жүйесіне арналған бірдей және бекітілген бөліктерге бөлінуін айтамыз.
- Жалпы жағдайда виртуалды адрестік кеңістіктің көлемі беттің көлемінен кіші болмайды, өйткені өр процестің соңғы беттері қосымша аумақпен толықтырылады.

- 
- Сонымен қоса ЭЕМ-нің барлық уақытша жадысы физикалық бет деп аталатын бөліктерге бөлінеді(*немесе блоктар немесе кадрлар немесе фреймдер*).
 - Әдетте беттің көлемі екінің дәрежесі ретінде таңдалады: 512, 1024 және т.б., бұл адрестерді өзгертудің механизмін жеңілдетуге мүмкіндік береді.
 - V әрбір кадр = V әрбір бет

Бос кадрлар арқылы беттерді тарату

Негізгі жады

Кадр Номер i	0	A.0	A.0	A.0	A.0	A.0
	1	A.1	A.1	A.1	A.1	A.1
	2	A.2	A.2	A.2	A.2	A.2
	3	A.3	A.3	A.3	A.3	A.3
	4		B.0	B.0		D.0
	5		B.1	B.1		D.1
	6		B.2	B.2		D.2
	7			C.0	C.0	C.0
	8			C.1	C.1	C.1
	9			C.2	C.2	C.2
	10			C.3	C.3	C.3
	11					D.3
	12					D.4
	13					
	14				29	

Жадыны беттік ұйымдастыру кезіндегі виртуалды адресті физикалық адреске өзгертуінің механизмі

- Онда блоктағы жеке элементті қалай табуға болады?
- Жеке блокты табу үшін сол элемент тұратын және блоктың басына қатысты элементті алтын блокты процесс көрсетуі қажет.
Адрестер келесі түрде болады (p, s) , мұндағы
 - p – процестің виртуалды бетінің номері (бетті номерлеу 0 ден басталады),
 - s - виртуалды бет шектерін алып тастау.

Виртуальный адрес

k двоичных разрядов

Номер виртуальной
страницы - p

Смещение в виртуальной
странице - S

№ вирт.
страницы

№ физич.
страницы

p

n

*Размер
страницы
 2^k*

Номер физической
страницы - n

Смещение в физической
странице - S

Физический адрес

k двоичных разрядов

- D процесі үшін орын жоқ. Бірақ ОЖ D процесін жүктейді.
- ОЖ ның әрбір процесі **беттік кестеден** тұрады
- Беттік кесте әрбір процестердің беттерінің кадрлерінің орнын көрсетеді.

0
1
2
3

-
-
-

7
8
9
10

4
5
6
11
12

13
14

Мысалға сәйкес мәліметтердің құрылымы (процестің кестесі А, В, С, D, бос кадрлердің тізімдері)

Екінші дәрежесіне тең көлемдегі беттер

- V әрбір кадр = V әрбір бет = $2k$

Сұрақтар

- **Логикалық, салыстырмалы және физикалық адрестердің айырмашылықтары қандай?**
- **Беттер мен кадрлар арасындағы айырмашылық?**
- **Сегменттер мен беттердің айырмашылығы?**
- **Жадыда келесідегідей ретпен, көлеммен орналасқан бос аймақтары бар тербеліс жүйесін қарастырамыз: 10 Кбайт, 4,20, 18, 7, 9, 12, 15 Кбайт. Осылардың қайсысы сегмент көлемінің сұранысын қанағаттандырады**
 - 12 Кбайт
 - 10 Кбайт
 - 9 Кбайт

Бірінші лайықты алгоритмі бойынша? Ең лайықты? Келесі лайықты?