

**O'zbekiston Respublikasi aloqa, axborotlashtirish va
telekommunikatsiya texnologiyalari davlat qo'mitasi**

Toshkent axborot texnologiyalari universiteti

Samarqand filiali

Axborot va pedagogik texnologiyalar fakulteti

Informatika va axborot texnologiyalari kafedrası

KURS ISHI

**Mavzu: Identifikatorlar jadvalini ro'yxat asosida
tashkillashtirish**

Bajaruvchi: 301 guruh talabasi

_____Mardiev U.

Kurs ishi rahbari:

_____Kat. o'qit. Murodov O'.

Samarqand – 2013

Мавзу: Идентификаторлар жадвалини рўйхат асосида ташкिलाштириш

Режа: 1. Ишнинг мақсади.

2. Идентифкаторлар жадвалининг тавсифи.
3. Идентификаторлар жадвалини ташкिलाштириш принциплари.
4. Идентификаторлар жадвалини қуришнинг рўйхат усули.
5. Ишни бажаришга қўйилган талаблар.
6. Топшириқнинг ечим босқичлари.
7. Хулоса.
8. Фойдаланилган адабиётлар.

1. Ишнинг мақсади.

Ишнинг мақсади идентификаторлар жадвалини ташкिलाштиришнинг асосий усуллари ўрганиш, идентификаторлар жадвалини ташкिलाштиришнинг турли усулларида келиб чиқадиган ютуқ ва камчиликлар ҳақида тасаввур ҳосил қилиш ҳисобланади.

Курс ишини бажариш учун киритишда идентификаторлар тўпламини оладиган, берилган усуллар ёрдамида идентификаторлар жадвалини ташкिलाштирадиган, жадвалларда хусусий идентификаторларни кўп бора излашга рухсат берадиган ва жадвални ташкिलाштиришнинг самарали усуллари солиштирадиган дастур тузиш талаб қилинади. Идентификаторлар рўйхати матнли файл кўринишида берилган деб ҳисобланади. Идентификатор узунлиги 32 белги билан чегараланган.

2. Идентифкаторлар жадвалининг тавсифи.

Семантик анализ, код генерацияси ва натижавий дастурни оптималлаштиришни бажаришда компилятор жорий дастур асосий

элементлари (ўзгарувчилар, ўзгармаслар, функциялар ва тилнинг бошқа лексик birlikлари) характеристикаларига боғланган бўлиши керак. Бу характеристикалар компиляторда кирувчи дастурнинг синтаксис анализи босқичларида олинади (кўпинча ўзгарувчи ва ўзгармасларни тасвирлашнинг блоклар структураси анализида) ҳамда код генерациясига тайёрланиш босқичида тўлдирилади (масалан, хотирани тақсимлашда).

Жорий дастур ҳар бир элементида мос тушадиган характеристикалар тўплами шу элемент типига, унинг маъносига (семантикасига) ва мос ҳолда жорий ва натижавий дастурлар бажарилиши ҳолига боғлиқ бўлади.

Ҳар бир аниқ ҳолда бу характеристикалар тўплами кирувчи тил синтаксиси ва семантикасига, бутун ҳисоблаш тизими архитектурасига боғлиқ бўлади. Аммо жорий дастур у ёки бу элементларига кўпинча мос тушадиган бир неча тур характеристикалар мавжуд. Масалан, ўзгарувчи учун – бу унинг тури ва хотира ячейкаси адреси, ўзгармас учун – унинг қиймати, функция учун – формал аргументлар сони ва турлари, қайтадиган натижалар турлари, функция кодини чақириш адреси. Жорий дастур характеристикалари ҳақида тўлиқ ахборотни, унинг анализини ва бажарилишини махсус адабиётлардан топиш мумкин.

Жорий дастур ихтиёрий элементининг бош характеристикаси унинг исми ҳисобланади.

Кирувчи тилнинг ўзгарувчилар, ўзгармаслар, функциялар ва бошқа элементлар исмлари билан дастур тузувчи шуғулланади – шунинг учун компилятор бу элементларни исмлари бўйича анализ қилиши керак. Ҳар бир элемент исми алоҳида кўринишга эга бўлиши керак. Кўпгина замонавий дастурлаштириш тиллари жорий дастурнинг кўринувчанлик соҳаси ва бошқа шартларга боғлиқ ҳолда ўзгарувчилар ва функциялар исмларида ўзаро ўхшаш бўлишига рухсат беради. Бу ҳолатда исмлар ягоналигини компиляторнинг ўзи таъминланиши керак, бу ерда жорий дастурнинг элементлари исмлари ягона ҳисобланади.

Шундай қилиб, компилятор вазифаси жорий дастурнинг ҳар элементиға боғланган қандайдир ахборотни сақлаш ва элемент исми бўйича бу ахборотга рухсатли кириш ҳисобланади. Бу масалани ечиш учун компилятор идентификатор идентификаторлар жадваллари ёки белгилар жадваллари деб аталадиган маълумотларнинг махсус сақланишини таъминлайди. Идентификаторлар жадвали жорий дастур битта элементиға мос тушадиган маълумотлар (ёзувлар) майдони тўпламидан иборат бўлади. Ёзув компиляторда берилган элемент ҳақида барча керакли ахборотни беради ва компилятор иши давомида тўлдирилиб бориши мумкин. Ёзувлар сони идентификаторлар жадвалини ташкиллаштириш усуллариға боғлиқ бўлади, аммо ҳар қандай ҳолатда жорий дастур элементларидан кам бўлади. Умуман олганда, компилятор бир эмас, балки бир нечта идентификаторлар жадваллари билан ишлайди – уларнинг сони ва структураси компилятор ишиға боғлиқ бўлади [1,2].

3. Идентификаторлар жадвалини ташкиллаштириш принциплари.

Компилятор идентификаторлар жадвалиға ёзувларни жорий дастур анализи ва жадвалда жойлашуви талаб қилинадиган янги элементларни топишиға кўра тўлдириб боради. Жадвалда ахборотни излаш ҳар доим компиляторға дастурнинг у ёки бу элементи ҳақида маълумот керак бўлганда бажарилади. Шунини эслатиб ўтиш керакки, компилятор билан жадвалда элементни излаш унга янги элементларни қўшишдан кўра кўпроқ бажарилади. Бундай бўлишға сабаб, жорий дастурда янги элементларни ишлатиш уларни тасвирлаш қондасиға кўра кам учрайди. Бундан ташқари, ҳар қандай ҳолатда идентификаторлар жадвалиға ҳар бир элементни қўшишда – шу элементни жадвалда мавжуд эмаслиғиға амин бўлиш учун излаш операцияси қатнашади. Жадвалда элементни излашнинг ҳар бир операциясида жорий дастурда элементлар сони катта бўлганлиғи (дастур ҳажмиға боғлиқ ҳолда бирдан то юз минггача) учун вақт сарфланади, бу эса вақт компиляциясининг умумий вақтиға таъсир этади. Шунинг учун

идентификаторлар шундай ташкилланиши керакки, унда компилятор бу ёзув боғланган элемент исми бўйича жадвалдаги унга керакли ёзувни излашни максимал тез бажариш имконига эга бўлиши керак.

Идентификаторлар жадвалини ташкиллаштиришнинг қуйидаги усуллари кўрсатиш мумкин:

- оддий ва тартибланган рўйхатлар;
- бинар дарахт,
- рехэширлаштириш билан хэш-адреслаш;
- занжир усули бўйича хэш-адреслаш;
- рўйхат ёки бинар дарахт билан хэш-адреслаш комбинацияси.

4. Идентификаторлар жадвалини қуришнинг рўйхат усули.

Оддий ҳолда идентификаторлар жадвали чизиқли тартибланмаган рўйхатни ёки ҳар бир ячейкаси жадвалнинг мос элементида маълумотлардан ташкил топган массивни тасвирлайди. Бу жадвалга янги элементни қўшиш массивнинг навбатдаги ячейкасига ахборотни ёзиш ёки жорий дастурда янги элементларни топиб рўйхат тузиш йўли билан бажарилади. Жадвалда керакли элементни излаш барча элементларни кетма – кет танлаш ва уларнинг исмларини шу исмли элемент топилгунча изланаётган элемент исми орқали таққослаш йўли билан бажарилади. У ҳолда агар вақт бирлигида иккита қаторни таққослаш учун компиляторда сарфланадиган вақт олинса (замонавий ҳисоблаш тизимларида бундай таққослаш кўпинча битта команда билан бажарилади), у ҳолда N элементдан ташкил топган жадвал учун ўртача $N/2$ таққослаш бажарилади.

Жадвалга янги элементни қўшишга (T_d) сарфланадиган вақт жадвалда элементлар сонига (N) боғлиқ эмас. Аммо агар N катта бўлса, у ҳолда излашга кўп вақт сарфланади. Бу жадвалдаги излаш вақтини $T_n=O(N)$ каби баҳолаш мумкин. Идентификаторлар жадвалида излаш компиляторда кўпинча бажариладиган операция ҳисобланади, идентификаторлар

жадвалини ташкиллаштиришнинг бундай усули самарасиз ҳисобланади. У кичик дастурлар билан ишлайдиган оддий компиляторларда қўлланилади.

Агар жадвалдаги элементлар табиий равишда тартибланган бўлса, у ҳолда излаш самарали бажарилади. Излаш исмлар бўйича амалга оширилганлиги сабабли, жадвалдаги элементларни тўғри ёки тескари алфавит тартибида жойлаштириш янади яхшироқ ечимга олиб келади. N элементлардан иборат тартибланган рўйхатда излашнинг самарали усули бинар ёки мантикий излаш ҳисобланади.

Мантикий излаш алгоритми қуйидагича бўлади: изланаётган белги жадвал ўртасида $(N+1)/2$ элемент билан таққосланади; агар бу элемент изланаётган бўлмаса, биз фақат 1 дан $(N+1)/2-1$ гача номерланган элементлар блокини ёки изланаётган элемент уни таққослаган элементдан катта ёки кичиклигига боғлиқ бўлмаган ҳолда $(N+1)/2+1$ дан N гача бўлган элементлар блокини қўришимиз керак.

Кейин жараён икки марта кичик ўлчамли керакли блокда такрорланади. Бу то изланаётган элемент топилмагунча ёки алгоритм битта ёки иккита элементдан (изланаётган элемент таққосланиши мумкин бўлган) ташкил топган кейинги блокка келгунча давом этади. Ҳар бир қадамда изланаётган элементдан ташкил топган элементлар сони 2 марта қисқарганлиги учун, таққослашларнинг максимал сони $1+\log_2 N$ га тенг. У ҳолда идентификаторлар жадвалида элементларни излаш вақтини $T_n=O(\log_2 N)$ каби баҳолаш мумкин. Таққослаш учун: бинар излашда $N=128$ учун энг катта таққослаш 8га тенг бўлса, тартибланмаган жадвалда излаш эса – ўртача таққослаш 64га тенг бўлади. Ҳар бир қадамда қўрилаётган ахборот ҳажми икки марта қисқарса, усул “бинар излаш” дейилади; массивда керакли элементни излашга кетадиган вақт ундаги элементларнинг умумий сонига мантикий боғлиқ бўлса, бу усул “мантикий излаш” дейилади.

Мантикий излашнинг камчилиги идентификаторлар жадвалини тартиблаш шарт ҳисобланади. Излаш бажарилаётган ахборот массиви тартибланган бўлиши керак, бунда уни тўлдириш вақти массивдаги

элементлар сонига боғлиқ бўлади. Идентификаторлар жадвали у тўлдирилгунча компиляторда қисмларда кўриб чиқилади, шунинг учун унга мурожаат қилишнинг барча босқичларида тартиблаш шarti бажарилиши талаб этилади. Шунга мувофиқ, бундай жадвални қуриш учун фақат элементларни қўшишнинг тўғри тартибланган алгоритмидан фойдаланиш мумкин.

Агар маълумотларнинг тартибланган массивларини ташкиллаштириш учун қўлланиладиган стандарт алгоритмлардан фойдаланилса, у ҳолда жадвалга барча элементларни жойлаштиришга кетадиган ўртача вақтни қуйидагича баҳолаш мумкин:

$$T_d = O(N \cdot \log_2 N) + k \cdot O(N^2)$$

Бу ерда k – таққослаш операцияси ва маълумотларни ўтказиш операциясини бажаришда компьютерда сарфланадиган вақтлар орасидаги муносабатни берадиган қандайдир коэффициент.

Идентификаторлар жадвалида мантиқий излашни ташкиллаштиришда жадвалга янги элементни қўшиш вақтини узайтириш ҳисобига керакли элементни излаш вақтини қисқартириш таъминланади. Идентификаторлар жадвалига янги элементларни қўшиш уларга мурожаат қилишга кўра кам амалга оширилганлиги учун, бу усулни тартибланмаган жадвални ташкиллаштириш усулига кўра самарали деб ҳисоблаш мумкин. Аммо реал компиляторларда унданда самарали усуллар борлиги учун бу усул ишлатилмайди.

5. Ишни бажаришга қўйилган талаблар.

Курс иши қуйидаги тартибда бажарилиши керак:

1. Талаба ўқитувчидан масала бўйича топшириқ олинади;
2. Идентификаторлар жадвалини ташкиллаштиришда берилган усуллар учун қўлланиладиган маълумотлар структурасини қўллаш амалга оширилади;

3. Ҳисобот тайёрланади ва ҳимоя қилинади;

Топширик вариантлари

1-жадвалда масалада қўлланиладиган идентификаторлар жадвалини ташкиллаштириш усуллари берилган.

1-жадвал. Идентификаторлар жадвалини ташкиллаштириш усуллари

Методлар №	Коллизияларга рухсат бериш усуллари
1	Оддий рехэширлаш
2	Тасодифий сонларни ишлатиш билан рехэширлаш
3	Кўпайтириш ёрдамида рехэширлаш
4	Занжир усули
5	Оддий рўйхат
6	Тартибланган рўйхат
7	Бинар дарахт

2-жадвалда 1-жадвалда берилган идентификаторлар жадвалини ташкиллаштириш усуллари асосида топшириқлар варианты варианты берилган.

2-жадвал. Масалалар вариантлари

Вариант №	Жадвални ташкиллаштириш- нинг биринчи усули	Жадвални ташкиллаштириш- нинг иккинчи усули
1	5	5

6. Топшириқнинг ечим босқичлари.

Дастурлашда фақатгина статик объектлардан фойдаланиш маълум бир қийинчиликларни, айниқса, самарали дастур тузишда, дастур самарали бўлиши, бир қатор амалий масаларни ҳал қилишда катта аҳамиятга эга. Сабаби, дастур тузилаётганда биз олдиндан объектларнинг нафақат қийматлари ўлчамини, балки мазкур объект дастурда иштирок этадими ёки йўқми шуни ҳам билмаслигимиз мумкин. Ушбу кўринишдаги дастур объектлари, яъни дастур бажарилиши жараёнида вужудга келадиган объектларга динамик объектлар дейилади.

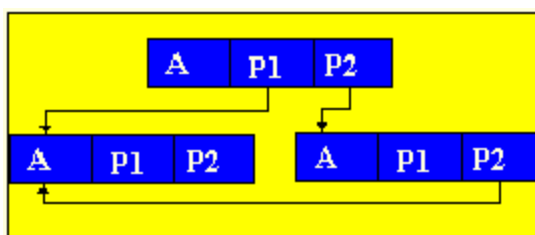
ПАСКАЛ тилида динамик объектлар билан ишлаш учун махсус маълумотлар турлари мавжуд. Ушбу турдаги маълумотлар қийматлари бирор бир дастурий объектга мурожаатни кўрсатади. Машина тилида бундай мурожаатлар мос объектни хотирадаги жойини намоёниш қилади.

Бундай ҳолатларда динамик объектлар устида амалларни бажариш учун ҳар бир объектга дастурда мурожаат туридаги статик ўзгарувчи мос қўйилади.

Динамик маълумотлар тузилмаси иккита ўзига хос хусусиятга эга:

1. Тузилмада элементлар сонини олдиндан аниқ айтиб бўлмайди;
2. Динамик тузилма элементлари қатъий чизиқли тартибланмаган. Улар хотирада тартибсиз равишда жойлашган бўлади.

Динамик тузилма элементларини ўзаро боғлаш учун элемент таркибига инфор­мацион майдондан ташқари кўрсаткичлар майдони киритилади (тузилма бошқа элементлари билан муносабатни ифодаловчи) .



p_1 , p_2 – берилган элементни тузилма бошқа элементлари билан алоқасини кўрсатувчи кўрсаткичлар.

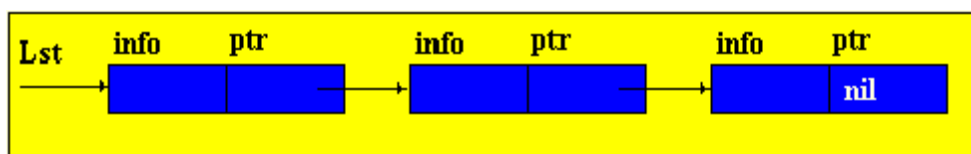
Энг кўп тарқалган динамик тузиламалар бўлиб боғламли рўйхатлар ҳисобланади. Рўйхатлар мантикий тасвирланиш нуқтаий назаридан иккига ажратилади: чизиқли ва чизиқсиз рўйхатлар. Чизиқли рўйхатларда боғлиқлик катъий тартибланган бўлади. Берилган элемент кўрсаткичи ўзидан кейин келувчи элемент адресини кўрсатади ёки аксинча.

Чизиқли рўйхатларга бир ва икки боғламли рўйхатлар киради.

Чизиқсизга эса кўп боғламли рўйхатлар мисол бўлади.

Умумий ҳолда рўйхат элементи ёзув (информацион майдон) ва бир ёки бир неча кўрсаткичлар майдони комбинациясини ифодалайди.

Чизиқли бир томонлама йўналган рўйхатлар



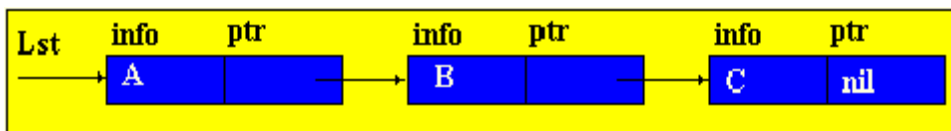
Бир боғламли рўйхат деб элементларни шундай тартибланган кетма-кетлигига айтиладики, ҳар бир элемент 2 та майдонга: информацион майдон *info* ва кўрсаткич майдони *ptr* га эга бўлади.

Мазкур кўринишдаги рўйхат элементи кўрсаткичини ўзига хослиги шундан иборатки, у фақатгина рўйхатнинг навбатдаги (ўзидан кейин келувчи) элементи адресини кўрсатади. Бир томонлама йўналтирилган рўйхатда энг сўнги элемент кўрсаткичи бўш, яъни *nil* бўлади.

Lst – рўйхат боши кўрсаткичи. У рўйхатни ягона бутун сифатида ифодалайди. Баъзан рўйхат бўш бўлиши ҳам мумкин, яъни рўйхатда битта ҳам элемент бўлмаслиги мумкин. Бу ҳолда $Lst = nil$ бўлади.

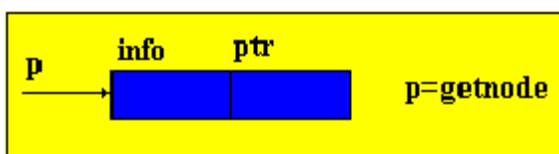
Бир боғламли рўйхатлар устида операциялар.

1. Бир боғламли рўйхат бошига элемент қўйиш.

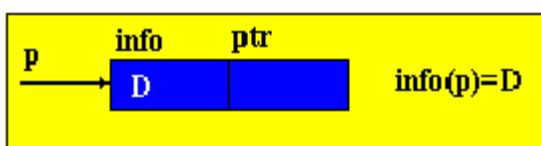


Берилган рўйхат бошига информацион майдони D ўзгарувчи бўлган элемент қўямиз. Ушбу ишни амалга ошириш учун қуйидаги ишларни бажариш лозим бўлади:

а) p кўрсаткич мурожаат қиладиган, бўш элемент яратиш.



б) Яратилган элемент информацион майдонига D ўзгарувчи қийматини ўзлаштириш.

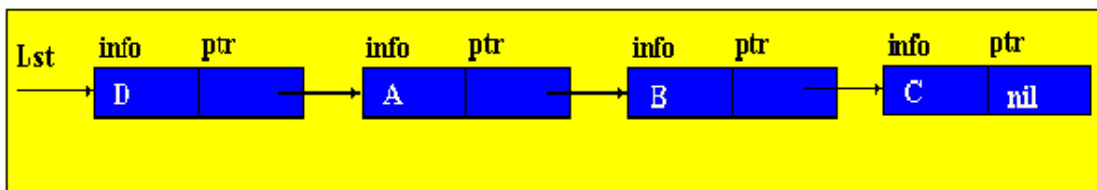


с) Янги элементни рўйхат билан боғлаш.

$\text{Ptr}(p)=\text{lst}$ (lst – энди рўйхат бошини кўрсатмайди)

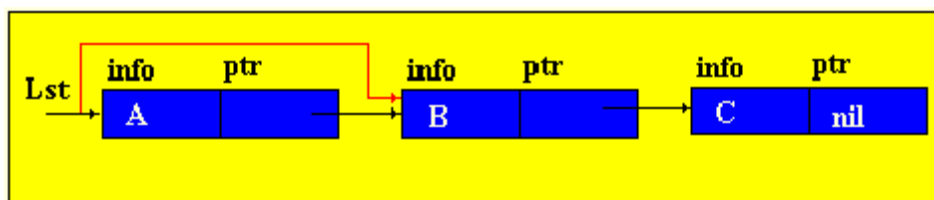
д) lst кўрсаткични рўйхат бошига кўчириш.

Ва ниҳоят:



Бир боғламли рўйхат бошидан элементни ўчириш

Рўйхатда биринчи элемент info информацион майдонидаги маълумотни эсда сақлаб қолиб уни рўйхатдан ўчираимиз.



Юқорида айтилганни амалга ошириш учун қуйидаги ишларни бажариш лозим:

a) Ўчирилаётган элементни кўрсатувчи p кўрсаткич киритиш.

$P = \text{lst}$

b) p кўрсаткич кўрсатаётган элемент info майдони қандайдир x ўзгарувчида сақлаш.

$X = \text{info}(P)$

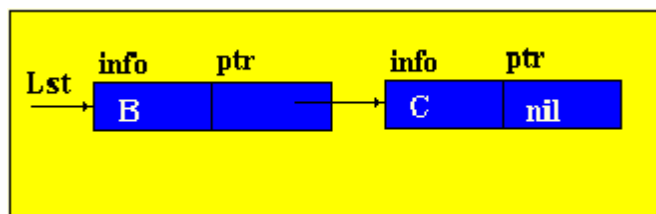
c) lst кўрсаткични янги рўйхат бошига кўчириш.

$\text{lst} = \text{ptr}(P)$

d) p кўрсаткич кўрсатаётган элементни ўчириш.

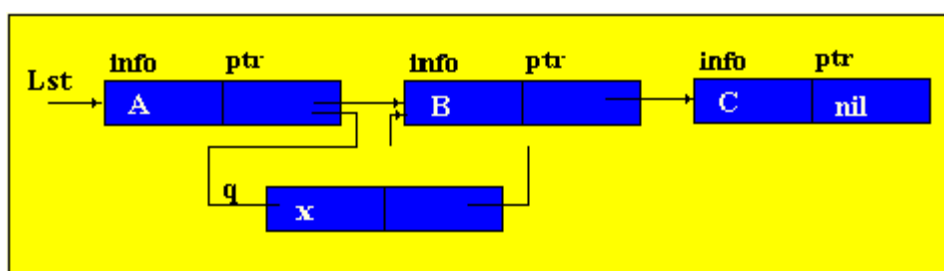
$\text{Freenode}(P)$

Натижада:



Элементни рўйхатга қўшиш

Берилган рўйхатда p кўрсаткич кўрсатаётган элементдан олдин
информацион майдони x бўлган элементни қўямиз.



Айтилганларни амалга ошириш учун қуйидаги ишларни бажариш
лозим:

a) q кўрсаткич кўрсатувчи бўш элемент яратиш

$q = \text{getnode}$

b) Яратилган элемент информацион майдонига x ни киритиш

$\text{info}(q) = x$

c) X элементни B элемент билан боғлаш

$\text{ptr}(q) = \text{ptr}(p)$ – яратилган элемент кўрсаткичига p элемент кўрсаткичини

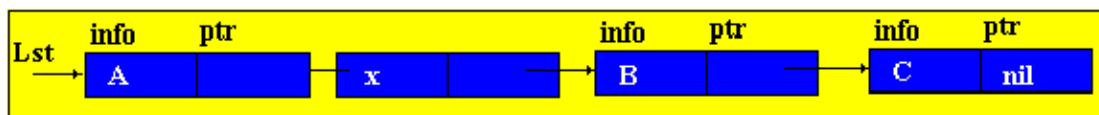
ўзлаштириш.

d) A элемент билан X элементни боғлаш

$\text{ptr}(p) = q$ – бу амал A элементдан кейинги элемент q кўрсаткич

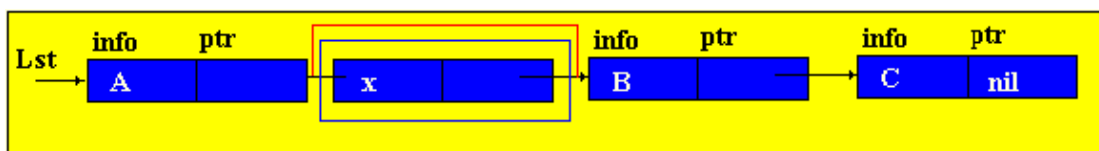
мурожаат қилган элемент бўлишини англатади.

Натижада:



Бир боғламли рўйхатдан элементни ўчириш

Рўйхатда ишчи кўрсаткичи p бўлган элементдан кейинги элементни ўчирамыз.



Буни рўёбга чиқариш учун қуйидаги ишларни амалга ошириш лозим:

а) Ўчирилаётган элементни кўрсатувчи q кўрсаткични киритиш.

$q = \text{ptr}(p)$

б) А элементдан кейин В элементни қўйиш

$\text{ptr}(p) = \text{ptr}(q)$

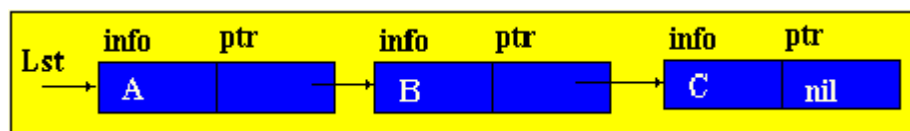
с) Ўчирилаётган элемент info мойдонидаги информацияни ёдда сақлаш.

$k = \text{info}(q)$

д) q кўрсаткичли элементни ўчириш.

$\text{Freenode}(q)$

Натижада:



**Бир ўлчамли массив кўринишида Паскаль алгоритмик тилида рўйхат
хосил қилиш дастури.**

```
const
```

```
    MaxQ = 17;
```

```
type
```

```
    E = [1..17] of char;
```

```
    Queue = Array [1..MaxQ] of E;
```

```
var
```

```
    Q: Queue;
```

```
    F, R: Integer;
```

```
{Кўрсаткич F рўйхат бошини кўрсатади. Кўрсаткич R эса, рўйхат охири  
кўрсатади. Агар рўйхат бўш бўлса, унда F = 1, акс ҳолда R = 0 (яъни R < F  
– рўйхатнинг бўш эканлиги шарти).}
```

```
Procedure Insert(I: Integer; var Q: Queue);
```

```
begin
```

```
    Inc(R);
```

```
    Q[R]:=I;
```

```
end;
```

```
Function Empty(Q: Queue): Boolean;
```

```
begin
```

```
    If R < F then Empty:=True Else Empty:=False;
```

```
end;
```

```
Procedure Remove(var I: Integer; Q: Queue);
```

```
begin
```

```
    If Not Empty(Q) then
```

```
begin
  I:=Q[F];
  Inc(F);
end;
end;
```


7. Х у л о с а.

Идентификаторларни ташкиллаштириш дастури компилятор таркибидаги асосий дастурлардан бири бўлиб, бошланғич дастурни таҳлил қилишда дастурда мавжуд бўлган идентификаторлар учун идентификаторлар жадвалини ҳосил қилади.

Идентификаторлар жадвалини ташкиллаштиришнинг қуйидаги усуллари кўрсатиш мумкин:

- оддий ва тартибланган рўйхатлар;
- бинар дарахт,
- рехэширлаштириш билан хэш-адреслаш;
- занжир усули бўйича хэш-адреслаш;
- рўйхат ёки бинар дарахт билан хэш-адреслаш комбинацияси.

Курс ишида ана шу усуллардан оддий ва тартибланган рўйхатлар усули ёрдамида идентификаторлар жадвалини ташкиллаштириш масаласи ўрганилган ва дастур тузилган.

8. Фойдаланилган адабиётлар.

8.1. Асосий адабиётлар.

1. Вирт Н. Алгоритмы и структуры программы. М., Мир, 1985.
2. Берзтисс А.Т. Структуры данных. М. Статистика, 1984.
3. Ленгсам Й. Структуры данных для персональных ЭВМ. М., Мир, 1989.

8.2. Қўшимча адабиётлар.

1. 6. Дж Ульман, и др. Введение в системы баз данных. М., Лори , 2000.
2. Д. Мейер, Теория реляционных баз данных, М, Мир, 1987.
3. Салахутдинов В.Х.. Конспект лекций по курсу “Структуры и организация данных в ЭВМ ” для студентов по специальности В-52200 “Информатика и информационные технологии” , Ташкент, 2000.

8.3. Информацион-техник воситалар.

Курс ишини бажариш учун компьютерлар билан жихозланган ўқув синфидан фойдаланилди.