2017

Т А Т У

Фарғона филиали

ТИЗИМЛИ ДАСТУРЛАШ МАЪРУЗАЛАР МАТНИ

Тузувчи: Дастурий инжиниринг кафедраси катта ўқитувчиси

Ф.Мулайдинов

“Дастурий инжиниринг” таълим йўналишларида таълим олаётган 3 курс ҳамда “Компьютер инжинирингги” ва “Телекоммуникация технологиялари ” таълим йўналишларида таълим олаётган 2 курс талабалари учун **«Тизимли дастурлаш»** фанидан маърузалар матни.

Маърузалар матни “Дастурий инжиниринг” кафедраси (“____” _____ 2016 йил, ___ - баённома), ва филиал услубий кенгаши (“____” _____ 2016 йил, ___ - баённома) йиғилишларида кўриб яиқилган ҳамда ўқув жараёнида фойдаланишга тавсия этилган.

Тақризчилар: _____ _____

_____ _____

_____ _____

_____ _____

“Дастурий инжиниринг”
кафедраси мудири: _____ **Х.Мусаев**

МУНДАРИЖА

Маъруза № 1. Мавзу: Тизимли дастурлаш асосий тушунчалари.

Режа:

1. **Тизимли дастурлаш асосий тушунчалари.**
2. **Тизимли дастурий таъминотнинг таркиби ва асосий функциялари.**

1.Тизимли дастурий таъминотнинг асосий тушунчалари

Калит сузлар.

- **Ҳисоблаш тизими**
- **Аппарат кисм**
- **Дастурий кисм**
- **Амалий дастурий таъминот**
- **Тизимли дастурий таъминот**
- **Умумий тизимли дастурий таъминот**
- **Махсус тизимли дастурий таъминот**

Замонавий амалиётда хисоблашларни бажариш учун хисоблаш машиналари ва тизимларидан фойдаланилади.

ЭХМ – бу аппаратуралар мажмуаси булиб, у кандайдир алгоритмлар тупламини харакатларини бажаришни таъминлайди.

Хисоблаш тизими – бу бир ёки бир неча ЭХМ ва дастурлар туплами булиб, узига юклатилган функцияларни бажарилишини таъминлайди.

Шундай килиб, хисоблаш тизимини 2 (икки) булакка булинади:

- дастурий;
- аппарат.

Аппарат кисми уз ичига хисоблаш машинасининг курилмаларининг бача функцияларини киритади. Аппарат кисмининг белгиланиши: киритиш ва чикариш амалларини бажариш, саклаш, узатиш ва маълумот алмашувни амалга оширишдан иборат.

Дастурий кисми эса – бу дастурлар туплами булиб, хисоблаш тизимига юклатилган функцияларни бажарилишини тартибини белгилайди. Дастур натижа олиш максадида аппаратура ишини бошкаради.

Дастурий булак, тизимли дастурий таъминот (ДТ) деб аталувчи бири бирига боглик купгина компоненталар тупламидан ташкил топган.

Изох: «таъминот» сузи хисоблаш тизимига юклатилган функцияларни амалга оширилишига курсатмадир.

Умуман олганда дастурий таъминот тизимини иккита катта булакка булинади:

1. Амалий ДТ (фойдаланувчи учун);
2. Тизимли ДТ.

Амалий дастурий таъминот – бу фойдалунувчиларнинг узлари учун узлари томонидан яратиладиган дастурлардир.

Тизимли дастурий таъминот – бу барча учун яратилган ва универсал булган дастурлардир. У хам икки булакка булинади.:

1. Умумий тизимли дастурий таъминот;
2. Махсус тизимли дастурий таъминот.

Махсус тизимли дастурий таъминот хисоблаш тизимининг аник специфик масалаларини ечиш учун умумий дастурий таъминотга кушилади (учишни бошкариш, харбий масалалар ва х.к.).

Умумий тизимли дастурий таъминот универсал булиб кенг омма вий масалаларни ечиш учун мулжалланган.

Бундан сунг умумий тизимли дастурий таъминотни куриб утамиз. У куйидаги таркибдан иборат:

1. тизимли кайта ишловчи дастурлар;
2. тизимли бошкарувчи дастурлар;
3. кушимча 1 ва 2 каби дастурлар;
4. текширувчи –диагностик дастурлар;
5. амалий дастурлар пакети;
6. Тизимли дастурий таъминот хужжатлари мажмуаси.

1. Тизимли кайта ишловчи дастурлар фойдаланувчиларга хизмат курсатиш масалаларини уларнинг талабларига кура ечишга мулжалланган.

2. Тизимли бошкарувчи дастурлар хисоблаш тизимининг барча функцияларини фойдалирок ташкил этиш ва хисоблаш тизими ва фойдаланувчи уртасидаги интерфейсни ташкил этиш учун мулжалланган.

Изох: Факатгина тизимли бошкарувчи дастурларгина аппаратурага бевосита мурожат кила оладилар.

Изох: Тизимли бошкарувчи дастурларни операцион тизимлар (ОТ) деб аталади.

ОТ интерфейснинг куйидаги вариантларини таъминлаши мумкин:

- команда интерфейси;
- дастур интерфейси (чакириклар тизими ёки баъзи бир тизимли функцияларни бажариш учун кисм дастурлар куринишида);
- фойдаланувчи интерфейси (дарча, меню, клавишалар ва х.к.)

3. Кушимча тизимли дастурлар кайта ишловчи ва бошкарувчи дастурларни имкониятларини кенгайтириш учун мулжалланган.

Улар таркибига :

- сервис дастурлари;
- инструменталь дастурлар киради.

Сервис дастурларга:

- дастур кобиклари (надстройки);
- утилиталар киради.

Дастур кобигининг яхши томони – бу хисоблаш тизимининг захираларига мурожатни яхшилашдан иборат (Windows проводниги в ах.к.).

Утилиталар (ёрдамчи хизмат курсатувчи дастурлар) фойдаланувчини кушимча имкониятлар билан таъминлаш (архивлаш, маълумотларни тиклаш, дискларга хизмат курсатиш, вирусга карши дастурлар).

Инструменталь дастурий курилмаларга куйидагилар киради:

- СУБД;
- Машина графикаси тизимлари

ва х.к.

4. Текширув-диагностик дастурлар ЭХМ нинг ишлатиш жараёнидаги носозликларни текшириш, аниклаш ва бартараф этиш профилактикаси учун мулжалланган.

5. Амалий дастурлар пакети – бу амалий масалаларни ечиш учун мулжалланган дастурлар тупламидир. Уларга – илмий хисоблашлар, моделлаштириш ва х.к. мисол булади.

Изох: Хар бир амалий дастурлар пакетининг уз тили булиб, бу пакетга тегишли ишларнинг бажарилиш тартиби ушбу тилда ифодаланади.

7. Хужжатлар мажмуаси – матнли хужжатларнинг ГОСТ (ЕСКД) га мувофик тайёрланган туплами булиб, уларда тизимли дастурий таъминотнинг мос булакларини эксплуатация килиш ва урнатиш хакидаги маълумотлар берилади.

2. Асосий функциялари ва таркиби

Тизимли кайта ишловчи дастурларнинг асосий функциялари ва таркиби;

1. Ассемблер;
2. Алока редакторлар ва юкловчилар;
3. Макропроцессорлар;
4. Трансляторлар (таржимонлар);
5. Тил конверторлари;
6. Редакторлар ва матн процессорлари;
7. Отладчиклар;
8. Дизассемблер;
9. Кросс-системлар;
10. Кутубхоначилар.

1. Ассемблер – бу шундай тизимли кайта ишловчи дастур булиб, у бирон бир машинага мулжалланган дастурлаш тилида ёзилган дастур матнини объект кодига айлантириш учун мулжалланган. (Ассемблер тилидаги матн директивалар ва исмлардан ташкил топади, машина коди эса факат байтлардан ташкил топади.).

Изох: Объект коди ёки алока редакторининг киришига, ёки юкловчининг киришига келиб тушади.

2. Алока редакторлари – бу шундай тизимли кайта ишловчи дастур булиб, улар Ассемблер ёрдамида алохида олинган объект модулларини ягона модулга бирлаштиришга мулжалланган. Алока редактори чегарасида барча адрес йуналишлари ягона адреслар фазосига урнатилади.

Алохида объект модулларида хар бир объект модули алока редакторининг чикишини юкловчининг кириши деб хисоблайди.

Юкловчилар дастурни кайта ишловчи дастурга юклайлилар ва унга бошкарувни узатадилар. Яна шу билан бирга улар юкловчиларни богловчи алохида модулларни бирлаштирадилар.

Изох: Юкловчилар жой узгартирувчи ёки абсолют булишлари мумкин.

- Абсолют юкловчилар хар бир дастурни биттадан фиксирланган адрес буйича юклайдилар.
- Жой узгартирувчи юкловчилар дастурни хотирадаги ихтиёрий буш жойга жойлаштиришлари мумкин.

3. Макропроцессорлар– бу шундай дастурларки, улар белгили кайта ишлашга мулжалланган булиб, бу жараёнда кандайдир киска фразаларга (макрочакирикларга) узун фразалар (макрокенгайтмалар) мос куйилади. Макропроцессорнинг киришида кандайдир макрочакириклардан олинган матн булиб , чикишида эса – макрокенгайтмалар булади.

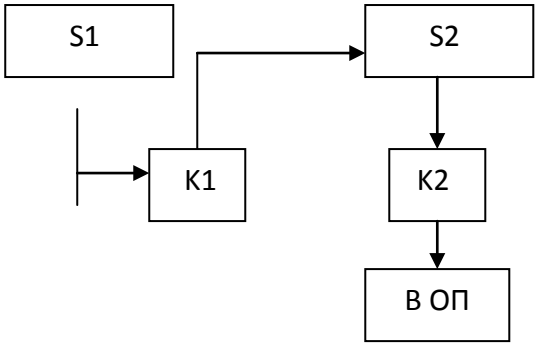
4. Трансляторлар (таржимонлар) бир тилда ёзилган матнни бошка тилга угирадилар. Трансляторларнинг куйидаги куринишларини ажратиб курсатиш мумкин:

- компиляторлар: киришида юкори даража тилида ёзилган дастур матни, чикишида машина кодларидаги алока редакторига ёки юкловчига узатиладиган дастур.

Хусусиятлари: таржима функцияларини аник булиниши ва таржима килинган функцияларни бажарилиши..

- интерпретаторлар – функциялар булинмайдилар, балки мослаштириладилар. Интерпретатор таржимани ва бажарилишни каторлаб ва кооператив бажаради. Улардан ёзилган дастурни диалог асосида кайта ишлашда фойдаланиш кулай.

5. Тил конверторлари бир юкори даража дастурлаш тилида ёзилган дастур матнини бошка юкори даража дастурлаш тилига айлантириш учун мулжалланган. Улар S1 дастурлаш тилида ёзилган дастурни S2 дастурлаш тилига айлантириш учун керак..



6. Матн редакторлари мантни кайта ишлаш учун кенг имкониятларга эгаликлари билан фаркланадилар.

7. Отладчиклар дастурларни бажарилиш жараёнида учраши мумкин булган хатоларни кидириш ва бартараф этиш учун мулжалланган.

8. Дизассемблер бу машина кодлари кетма-кетлигини ассемблер куринишига узгартирадиган дастур.

Изох: Улар хам баъзи бир харакатлар тизимини бажарилишини ассемблер куринишида куриш имконини яратадилар.

9. Кросс-тизим – бу дастур бир хисоблаш машинасида машина кодларида ифодаланган бошка бир хисоблаш машинасининг дастурларини олиш учун кулланилади. Лойихалаштирилаётган хисоблаш тизимлари архитектурасини отладка килиш учун фойдаланилади.

10. Кутубхоначилар – киритилаётган матн , объект модули раками булиши мумкин булган кутубхона файлларини ташкил этиш ва уларга хизмат курсатиш учун дастурлардир.

Операцион тизим ривожланиш имкониятлари:

Операцион тизим узлуксиз ривожланишининг сабаблари:

1. Аппарат таъминотининг янги куринишларининг юзага келиши ва янгиланиши.

2. Янги сервислар.

3. Узгартиришлар киритиш.

Операцион тизимнинг ривожланиши зарурият, акс холда у янги дастурлар ва янги курулмалар билан ишлай олмайди.

Мисол учун, Intel процессорлар Hyper Thread ни кулланишидан бошлаб Windows NT операцион тизимини бозордан сикиб чикара бошлади, улар иккинчи процессорни эмуляция килдилар. Как работать с эмуляцией ядра Windows NT ни ядросини эмуляцияси билан кандай ишлашни «билмай колиб» тизим йикилди.

Операцион тизимлар ривожланиши мумкин.

1.сакраб (Windows) ва
2.аста секин (UNIX).

Биринчи холда аввалги мавжуд дастурлардан тулик воз кечишга тугри келади, чунки бу тизимда хеч кандай кучириб утишлар назарда тутилмаган. Аммо, факат унинг узигагина ёзилган янги дастурлардан бутун махсулот юзага келади. Бу эса дастурий таъминотни ишлаб чикарувчилар (яратувчилар) учун жуда фойдалидир.

Иккинчи холда янги курилма имкониятларини эски дастурлар билан боглашга тугри келади. Операцион тизим ва дастурлар имкониятларини кенгайтирувчи дастур заплаткаларини (патчлар) ишлаб чикаришга тугри келади.

Операцион тизимларни ишлаб чикиш.

Операцион тизимларни ишлаб чикиш буйича уч ечиш йули мавжуд.

Ёпик ечим. Барча ишлаб чикарилаётган махсулотлар лицензиялар билан химояланган ва ишлаб чикарилган фирмадан ташкарида узгартирила олмайди. Бундай ечим фирмадан курилмаларнинг узгаришига жуда тез эътибор беришни, маркетинг ва ишлаб чикаришга катта маблагларни сарфлашни талаб килади.

Бундай йуналишнинг ютуги булиб, узи ишлаб чикараётган дастурий махсулоти учун фирманинг жавобгарлиги хисобланади. Яхши хужжатлаштириш. Универсальлик. Стандартлаштириш. Бундай йуналишдан фойдаланаётган таникли фирмалардан : Microsoft, SUN, SCO.

Бундай йуналишнинг камчилиги фирмаларнинг инертлиги хисобланади. Узгараётган шароитларга тезда ахамият бера олмаслик кобилияти. Операцион тизимнинг бошка фирма ишлаб чикараётган операцион тизим билан узаро алока кила олмаслиги, яъни мослаша олмаслиги имконияти.

Очик ечим. Ишлаб чикарилаётган ечимлар умумий лицензияга эга очик кодли ечимга буйсунадилар. Ихтиёрий хохловчи дастур махсулотининг берилиш кодини олиши мумкин, агар унинг автор томонидан ишчи варианти куйилган булса. Бундай ечим дастурий махсулот хатосиз ишлашига кафолат бера олмайди. Бу ечимлар куп холларда яхши хужжатлаштирилмаган. Бундай йуналишнинг ютуги булиб, турли давлат ва турли соха мутахассисларининг бир командада ишлай олиши имконияти хисобланади. Ишлаб чикарувчилар командасининг узгариш шартларига тезликдаги эътибори. Барча мумкин булган платформаларда ва ихтиёрий бошка тизимларда ишлай олиш имконияти. Бундай йуналишдан фойдаланаётган таникли фирмалардан: RedHat, SuSe, FreeBSD; Novell.

Аралаш ечим. Ишлаб чикарилаётган ечимлар умумий коддан ташкари яна шахсий ишлаб чикарилаётган, лицензия билан химояланган ечимдан хам фойдаланадилар. Бундай йуналиш очик ечимлардан энг яхши ечимларни танлаб олиш ва улар асосида узларининг ечимларини ташкил этиш имконини беради. Бундай йуналиш хар иккала йуналишнинг энг яхши хусусиятларини хисобга олади, чунки фирма факатгина узининг ечимларинигина хужжатлаштириш ва унга жавобгарликни буйнига олибгина колмай, балки танлаб олинган очик ечимлар учун хам жавобгарликни хис этади. Бундай йулни танлаган фирмалар MacOS, BeOs, QNX, Netrino.

Назорат саволлари

1. Хисоблаш тизимини кандай булаклар ташкил этади?
2. Амалий дастурлашнинг вазифаси нималардан иборат?
3. Тизимли дастурлашнинг вазифаси хакида тушунча беринг**.**
4. Фойдаланувчи интерфейси нима учун керак?
5. Дастур кобиклари нима?
6. Утилиталар нима?
7. Тизимли кайта ишловчи дастурларнинг асосий функциялари ва таркиби хакида маълумот беринг.

Фойдаланилган адабиётлар

1.Афанасьев А.Н. Формальные языки и грамматики: Учебная школа: УлГТУ, 1997. – 84 с

2.Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции -: Мир, 1979.-487с.

3.Компаниец Р.И. Системное программирование. Основы построения трансляторов. СПб.:Корна принт., 2000. -256 стр.

4.Дьяконов В.Ю. Системное программирование. Высш.шк.. 1990. -221 с.

Маъруза № 2. Мавзу. Операцион тизимлари. Файл ва файллар тизими.

Режа:

1.Файлларнинг асосий хусусиятлари.

2.Берилганларни химоялаш.

3.Файллар тизимининг асосий хусусиятлари.

Калит сузлар.

- **Файл**
- **Файллар тизими**
- **Файл кўрсаткичи**
- **Шахсийлаштириш**
- **Операцион тизим**
- **Узлуксиз сегментли файллар**
- **Блокли ташкил этилган файллар тизими**
- **Иерархик файллар тизими**

Биз аввал айтиб утганимиздек, хар бир операцион тизим жараён деб аталувчи тушунчага эга. Яна иккинчи бир тушунча хам борки, у хам жуда мухим булиб –бу файлдир. *Файл тизими - бу операцион тизим компонентаси булиб, номланган берилганлар тупламига мурожаатни, ташкил этиш ва саклашни таъминловчидир.Берилганларнинг номланган туплами файллар деб аталади.*

1.Файлларнинг асосий хусусиятлари.

1. Файл -бу исмга эга объект булиб, шу исм оркали файлни ичидаги маълумотлар билан ишловчи объектдир. Исм бу белгилар кетма-кетлиги булиб, унинг узунлиги аник операцион тизим турига богликдир.

2. Файлни жойлашишига боглик эмаслиги. Аник бир файл билан ишлаш учун у файлнинг ташки курилмадаги жойлашишини билиш талаб килинмайди.

3. Кириш/чикиш функциялари туплами. Хар бир операцион тизим файллар билан маълумот алмашинувни таъминловчи функциялар тупламига эга. Бу функциялар туплами куйидагилардан ташкил топади:

1. Файл иш учун очилган. Ёки мавжуд ёки янги файлни очиш мумкин. Шундай савол тугилиши мумкин. - нима учун файлни очиш керак? Нима учун бирданига файлдан укиш ва файлга ёзиш мумкин эмас? Хакикатда, бу операцион тизимга файл аник жараён билан ишлашини марказий равишда эълон килиш воситасидир. У эса ушбу маълумотларга асосан кандайдир ечим кабул килиши мумкин. (масалан, бошка жараёнлар учун ушбу файлга мурожаатни чеклаб куйиши мумкин.).

2. Укиш/ёзиш. Купинча файл билан маълумот алмашинув берилганлар блоки куринишида ташкил этилиши мумкин. Ушбу берилганлар блоки икки хил хусусиятга эга. Бир томондан ихтиёрий хисоблаш тизими учун берилганлар блокининг улчами аник берилган, яъни булар аппарат -дастур улчамларидир. Иккинчи томондан бу берилганлар блоки реал алмашинувда дастурчи томонидан ихтиёрий равишда бошкарилиши мумкин. Укиш/ёзиш функцияларида купинча алмашинув учун берилганлар блоки улчами ва укилиши ёки ёзилиши керак булган берилганлар блоки сони берилади. Танланган берилганлар блокининг улчамидан алмашинувларнинг унумдорлиги боглик, фараз килайлик бир машина учун берилганлар блокининг унумдорлик улчами 256 Кб булса, сиз 128 Кб лик алмашинувни амалга оширмокчи булсангиз, у холда сиз мантикий блокларни укиш учун икки маротаба 128 Кб дан мурожаат киласиз. Бу холда сиз 256 Кб ни бир мартада укиш урнига, бир блокга икки маротаба мурожаат киласиз ва бир сафар ярмини, кейинги сафар кейинги ярмини укийсиз. Бу ерда яна баъзи бир

«унимсизлик» элементлари учраши хам мумкин, лекин уларни «аклли» операцион тизим текислаб юборади, агар текислай олмаса, демак бу сизнинг хатойингиз булади.

3. Файл курсаткичини бошкариш. Хар бир очилган файл билан файл курсаткичи тушунчаси боглик. Бу курсаткич командаларнинг хисоблагич регистри булиб, хар бир вактда кейинги файл буйича алмашинувни амалга ошириш мумкин булган нисбий адресни курсатади. Ушбу блок билан алмашинув тугагандан сунг курсаткич блокдан ташкарига кучирилади. Файл билан ишни ташкил этиш учун ушбу файл курсаткичини бошкариш талаб этилади. Файл курсаткичини бошкариш функцияси мавжуд булиб, курсаткични файл буйича ихтиёрий (мумкин булган чегараларда) кучириш имконини беради. Курсаткич бу кандайдир узгарувчи булиб, дастурдан мурожат килиш мумкин, ва у файлни очиш функцияси (ушбу узгарувчини ташкил этувчи) билан боглик.

4. Файлни ёпиш. Бу амал иккита функция оркали амалга оширилиши мумкин:

1) Файлни ёпиш ва охирги кийматини саклаб колиш.

2) Файлни йукотиб (учириб) ташлаш.

Файл ёпилгандан сунг у билан барча алокалар тугатилади ва у каноник холатга утади.

2. Берилганларни химоялаш. Купгина стратегик ечимлар аппарат даражасидаги каби, операцион тизим даражасида хам такрорланади. Агар биз мультипрограмма режимини эласак, унинг мавжуд булишлигининг зарурий шартларидан бири бу хотира ва берилганларни химоялашни таъминлашдан иборат эди. Агар биз файллар тизимини карасак, у хам худи операцион тизим каби бир фойдаланувчилик булиши мумкин. Бу холатда берилганларни химоялаш муаммоси булмайди, чунки операцион тизимда ишлаётган киши барча файлларни эгаси хисобланади. MS-DOS ёки Windows бир фойдаланувчилик тизимларга мисол булади. Машинани юклаб бошка фойдаланувчиларнинг барча дисклардаги файлларини учириб ташлаш мумкин, чунки бу тизимларда хеч кандай химоялаш йук. Куп фойдаланувчилик тизим купгина фойдаланувчиларнинг тиник ишлашини таъминлайди. MS-DOS операцион тизими хам мультипрограмма режимида ишлаши мумкин, лекин жуда хам тиник эмас, бир жараёндаги хатолик операцион тизимнинг ва кушни жараённинг учирилишига олиб келиши мумкин. Худди шундай Windows операцион тизимида хам купфойдаланувчи ишлаши мумкин, лекин унинг иши хам тиник эмас, чунки операцион тизим уланрнинг барча хукукларини химоясини таъминламайди. Шундай килиб, купфойдаланувчилик тизим ташки таъсирлардан химоя килиши керак. Аслида химоялаш муаммоси факатгина файл тизими билан боглик эмас. ОТ барча сохаларда берилганларни химоялашни таъминлайди: бу файллар, жараёнлар, жараёнларга тегишли бир фойдаланувчи томонидан куйилган ресурслар. Бу ерда мен сизнинг эътиборингизни шу фактга каратаман, чунки файллар учун бу жуда мухим нукта.

3.Файллар тизимининг асосий хусусиятлари.

Файллар тизими файллар учун санаб утилган барча хусусиятларни уз ичига олади, ва яна баъзи бир кушимча хусусиятларга хам эга. Бу хусусиятлар файллар тизимининг структуралик ташкил этилиши билан боглик.

Келинг, кандайдир ташки саклаш курилмаси (ВЗУ) фазосини караб чикайлик ва бу фазода файлларни жойлаштиришни ташкил этишни куриб чикамиз.

1. Узлуксиз сегментли файлларни бирдаражали ташкил этиш. «Бирдаражали» **т**ермини тизим уникал номланган файллар билан ишлашни таъминлашини англатади. Ташки саклаш курилмаси чегарасида берилганларни саклаш учун каталог деб аталувчи соха ажратилади. Каталог куйидаги структурага эга:

исм	Бошлангич блок	Охирги блок

«Бошлангич блок» берилган исм билан бошланувчи ташки саклаш курилмасидаги нисбий адресни курсатади. «Охирги блок» берилган файлнинг охирги блокини аниклайди. Файлни очиш функцияси каталогда файл исмини топиш, унинг бошланиши ва охирини аниклашни амалга оширади. (амалда берилганлар курсатилганидан кам жой эгаллаши мумкин, лекин бу хакда кейинрок тухталамиз). Бу харакат жуда оддий, шу билан каталогни ОТ хотирасида саклаш мумкин, бу эса алмашинувларни камайишига олиб келади. Агар янги файл ташкил этилаётган булса, у буш жойга ёзилади. Исмлар каталогига ухшаш буш фазолар (фрагментлар) жадвали булиши мумкин.

Укиш/ёзиш кушимча алмашинувларсиз амалга оширилади, чунки файлни очишда биз берилганларни жойлаштириш диапазонига эга буламиз. Укиш ушбу структура блокига мос равишда амалга оширилади ва хеч кандай кушимча маълумот талаб этилмайли, алмашинув хам мос равишда тезда амалга оширилади.

Энди караб чикайлик, бундай файлга кушимча маълумот ёзмокчимиз, лекин буш фазо жой йук? Бу холда тизим икки хил йул тутиши мумкин. Биринчидан, у сизга жой йуклигини айтади ва сиз узингиз нимадир килишингиз керак булади, яъни кандайдир Ушбу файлни бирор жойга кучириб турадиган ва керакли маълумотни кушадиган жараённи куясиз. Бундай кучириш етарли даражада кимматга тушадиган функция. Иккинчи имконият - алмашинувни рад этилади. Бу эса файлни очиш жараёнида аввалдан кушимча жой олиб куйиш кераклигини англатади; бу холда файл тизими буфернинг буш улчамини текширади ва у кам булса, Ушбу файлни жойлаштириш учун буш жой кидиради.

Бундай курамизки, бундай ташкил этиш сода, алмашинувларда унумли, лекин файл учун жой етишмаган холларда унимсизлик бошланади. Бундай ташкари файл тизимининг узок ишлаши давомида дискда худди оператив хотирадаги каби фрагментация холати юз беради. Яъни буш жойлар мавжуд ,лекин файлимизни жойлаштириш учун етарли жой йук булган холат юзага келади. Файл тизимини бундай ташкил этилишининг фрагментацияси билан курашишда узок, огир ва файл тизими учун хавфли булган жараён, яъни файлларни бир-бирига зичлаштириш жараёни амалга оширилади.

Бундай ташкил этиш бир фойдаланувчилик файл тизими учун кулай ва фойдалидир, чунки фойдаланувчиларнинг куплиги холатида фрагментация юз беради. Зичлаштириш жараёнини хар доим куйиш максадга мувофик эмас. Бошка томондан тизим оддий ва хеч кандай кушимча харажатлар талаб килмайди.

2. Файллар блокли ташкил этилган файллар тизими. Ташки саклаш курилмалари фазоси блокларга булинган. Файллар тизимида бундай маълумотларни булаклашда оператив хотирани варакли ташкил этишдаги жараёнлар маълумотларини булаклаш каби амалга оширилади. Умумий холда, хар бир файл исми билан шу файл берилганлари жойлашган курилма блоклари ракамларини туплами боглик. Ушбу блокларни ракамлари ихтиёрий тартибга эга, яъни блоклар курилма буйича ихтиёрий таркалган. Бундай ташкил этишда фрагментация муаммоси йук, аммо блокни яхлитлаш йукотишлари мавжуд (агар файл блокни битта байтини банд килган булса, у холда бутун блок банд хисобланади). Шундай килиб, зичлаштириш муаммоси йук, ва бу тизим купфойдаланувлилик ташкил этишда фойдаланиш мумкин.

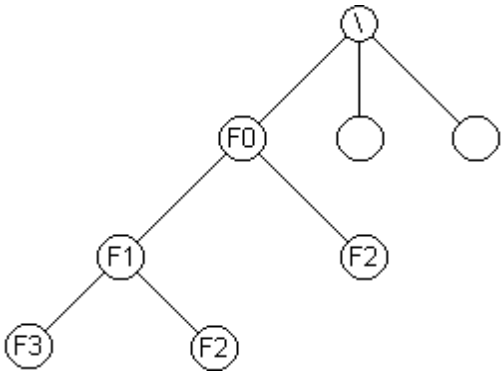
Бу холда хар бир файл бир канча атрибутлар туплами билан боглик: файл исми ва файлга мурожаат этиладиган фойдаланувчи исми; Бундай ташкил этилиши исмларни уникаллиги муаммосидан кутилишга имкон беради. Бундай тизимда исм уникаллиги бир фойдаланувчилик файллар уртасида талаб этилади.

Бундай файлларни ташкил этиш каталог оркали амалга оширилади. Каталогни структураси куйидагича булади. Каталог каторлардан ташкил топади; хар бир і-чи катор файл тизимини і-чи блокга мос келади. Бу каторда блок банд ёки бушлиги хакидаги маълумот сакланади. Агар у банд булса, у холда бу каторда файл исми (ёки унга мурожаат), фойдаланувчи исми ва бошка кушимча маълумотлар жойлашиши мумкин.

Маълумот алмашинув даврида тизим турлича харакатланиши мумкин. Ёки файлни очишда тизим каталог буйича айланиб файлни мантикий блокларини дискда жойлашиш жадвалини куради. Ёки хар бир алмашинувда бу мослик амалга оширилади.

Файллар тизимини бундай ташкил этилиши бир фойдаланувчи рамкасида бирдаражали хисобланади, яъни барча файллар кандайдир фойдаланувчига тегишли гурухга богланган.

3. Иерархик файллар тизими. Файл тизимининг барча файллари дарахт деб аталган бир структурага курилган. Дарахтнинг илдизида файл тизимининг илдизи жойлашган. Дарахтни богланган жойи варак хисобланса, бу файл фойдаланувчининг берилганларидан ташкил топиб файл-каталог хисобланади. Дарахтнинг варакдан фаркли богланишлари файл-каталоглар хисобланади. Бундай иерархик файл тизимида номланиш турли усуллар билан амалга ошади.



Биринчи тур – файлни энг якин каталога нисбатан номлаш, яъни биз F0 каталог учун якин булган файлларни карасак, бу F1 файл булиб, у хам каталогдир ва F2 файл. Бундай тизимда бир даражада исмлар такрорланмаслиги максадга мувофик. Бошка томондан, барча файллар дарахт билан богланганликлари учун биз файлни тулик номи, яъни файл тизими илдизидан аник бир файлгача булган йул, хакида гапира оламиз. F3 файлнинг тулик исми куйидагича белгиланади: /F0/F1/F3. Бундай ташкил этиш файлнинг киска исми билан хам, тулик исми билан хам ишлаш имконини беради. Файлларнинг тулик исми бу йулдир. Ихтиёрий дарахтда унинг илдизидан ихтиёрий богламигача бита йул мавжуд, шундай килиб исмларни унификация килиш муаммоси хал этилади. Биринчи марта бундай усул Беркли университетида 60-йилларнинг охирида ишлаб чикилган Multix операцион тизимида фойдаланилган. Кейинчалик бу чиройли ечим купгина операцион тизимларда кулланила бошлади. Бу иерархияга мос равишда хар бир файлга кандайдир мурожат хукукига эга атрибутларни боглаб куйиш мумкин. Мурожаат хукукларига фойдаланувчилар файллари билан биргаликда каталоглар хам эгадирлар.Бу тизимнинг структураси купфойдаланувчилик ишни ташкил этишда, исмлар муаммосини йуклиги хисобига унумдордир.

4. Операцион тизимда шахсийлаштириш ва берилганларни химоялаш. Бу нюанс, хам сода, хам мураккаб. Соддалиги шундаки биз у хакида бир неча огиз гапирамиз холос, мураккаблиги шундаки, шундай муаммолар мавжудки улар хакида узок гапириш мумкин.

Шахсийлаштириш – бу аник фойдаланувчини идентификация кили шва шу билан мос равишда берилганларни химоялаш буйича у ёки бу харакатларни кабул килиш имкониятидир.

Агар биз ихтиёрий MS-DOS операцион тизимини карасак, у бир фойдаланувчилик.

Операцион тизимларнинг иккинчи даражаси – фойдаланувчиларни руйхатдан утказадиган, лекин барча фойдаланувчилар ягона субъект туплами куринишида ва бир-бири билан боглик эмаслар. Бундай операцион тизимларга мисол сифатида IBM фирмасининг mainframe-компьютерлари учун операцион тизимларини курсатиш мумкин. Мисол учун маърузачи узининг эшитувчи талабаларнинг кайси бири кандай гурухга тегишли эканлигини билмайди, лекин уларнинг шу крс талабалари эканлигини билади. Бу хам яхши, хам ёмон. Бу курсни эшитиш учун яхши, лекин маърузачи томонидан савол жавоб килиш масаласида ёмон, чунки бир кун ичида у хамма талабалар билна савол-жавоб килишга улгура олмайди. Шунинг учун у барча эшитувчиларни кандайдир булаклаши керак, лекин кандай бу ноаник.

Мос равишда бундай бир улчамли шахсийлаштиришда барча биз айтиб утган функциялар (хусусан, химоялаш) таъминланади, лекин фойдаланувчиларни бундай ташкил этиш фойдаланувчилар гурухини англатмайди. Менга эса бизнинг факультет серверида мениг лабораториям ажратилса ва бу лаборатория рамкасида файлларга мурожаат хукукини бир бирига бериш имкони берилса максадга мувофик булар эди.

Мос равишда файллар тизимидаги каби фойдаланувчиларнинг иерархик ташкил этиш пайди булади. Яъни бизда «барча фойдаланувчилар» ва «фойдаланувчилар гурухи» деган тушунча мавжуд. Гурухда реал фойдаланувчилар мавжуд. Бундай шахсийлаштиришни иерархик ташкил этиш куйидаги ларни келтириб чикаради. Кандайдир фойдаланувчини руйхатдан утказиш учун уни аввал кандайдир гурухга киритиш керак, - бу лаборатория, кафедра ёки укув булимии булиши мумкин. Фойдаланувчилар гурухларга бирлашганликлари учун, фойдаланувчиларнинг ресурсларига мурожаат хукукини булиниш имконияти юзага келади. Яъни, масалан фойдаланувчи унинг ресурсларидан барча гурухдаги фойдаланувчилар фойдаланишлари мумкин эканлигини эълон килиши мумкин. Бундай чизма купдаражали (гурухлар гурухчаларга булинадилар ва х.к.) мос хукук ва имкониятлардан келиб чиккан холда булиши мумкин. Хозирги кунда шундай операцион тизимлар яратилмокдаки, уларда мурожаат хукуки факатгина иерархик структурага боглик булиб колмай, балки мураккаброкдир, яъни мурожаат хукукини иерархияни бузган холда кушиш мумкин.

Назорат саволлари.

1. Файл нима ?
2. Файлларнинг асосий хусусиятлари хакида маълумот беринг.
3. Файл курсаткичи нима?
4. Файллар тизими нима ?
5. Файллар тизимининг асосий хусусиятлари хакида маълумот беринг.
6. Узлуксиз сегментли файлларни бирдаражали ташкил этишнинг ютук ва камчиликлари хакида маълумот беринг.
7. Файллар блокли ташкил этилган файллар тизимининг ютук ва камчиликлари хакида маълумот беринг.
8. Иерархик файллар тизими тизимининг ютук ва камчиликлари хакида маълумот беринг.

Фойдаланилган адабиётлар

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. –СПб: Питер, 2003.- 396 с.
2. Афанасьев А.Н. Формальные языки и грамматики: Учебная школа: УлГТУ, 1997. – 84 с
3. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции -: Мир, 1979.- 487с.
4. Компаниец Р.И. Системное программирование. Основы построения трансляторов. СПб.:Корна принт., 2000. -256 стр.
5. Дьяконов В.Ю. Системное программирование. Высш.шк.. 1990. -221 с.

Маъруза №3. Мавзу: Дастурлаш таъминот ва уларнинг классификацияси.

РЕЖА

1. Дастурлаш фанининг тарихий боскичлари.
2. ДТ ишончлилиги.
3. Дастурлаш- маълумотларни кайта ишлаш жараёнини формал ёритиш сифатида.
4. Тугри дастур тушунчасининг ноконструктивлиги

Таянч сўзлар: Продседура ишлатиш, дастурлаш кутубхонаси , Структураланган дастур, COM (component object model), CORBA (common object request broket architecture), CASE технологияси, CASE-Computer aided Softwore/system Engineting

.

Дастурлаш фанининг тарихий боскичлари

1- Боскич стихик программа

Бу боскич натижалари
Продседура ишлатиш, дастурлаш кутубхонаси. 1- боскич дастурлаш кутубхонасини тузиш.

2- боскич. Структураланган дастур – бу усул асосида декомпозитция тушунчаси туради. Яни ката хжмдаги дастурни бир неча кичик дастурларга булиш. Модуллараро богланиш махсус интерфейс оркали бажарилади. Бу технология Nurbo Pasksl, C++, AD ва бошка тилларда ишлатилади.

3- Боскич. Обектга мулжалланган дастурлаш технологияси. Бу технология асосида куйидагилар: обйект, хабар, компонента, синфлар киради. Дастур бир неча абектдан иборат булиб, хар бир обект алохида синфнинг экземпляридир. Шу етапда визуал дастурлаш технологияси хам пайдо булган.

4- Боскич. Компонентали дастурлаш ва CACE технологиялари. Компонентали йуналиш асосида дастурлаш таъминотни алохида компоненталардан яратиш усуллари назарда тутилади. Компонента – бу динамик равишда чакирилаётган дастурлар кутубхонаси. Шу кутубхоналарни ёки бажарилаётган файлларни стандарт 2лик куринишида ёзиш мумкин. Компонентали йуналиш куйидаги технологиялар асосида ишлатилади:

COM (component object model)
CORBA (common object request broket architecture)
CASE технологияси структураланган обектга мулжалланган компонентали технологияларни уз ичига олади.
CASE-Computer aided Softwore/system Engineting.

Дастурий таъминотнинг ишончлиги

Дастурий махсулот фойдаланувчи учун зарур бўлган барча ахборотни бера олмас экан, бу махсулотни ишончли деб бўлмайди ёки бу холда дастурй махсулотда хатолик мавжуд дейилади. Дастурий таъминотдаги хатоликлар, унинг ички хусусиятлари бўлиб хисобланмайди. Бу эса программавий таъминотда махсулотни канча кўп тестланса у шунча яхши ишлайди деган фикр инкор килади. Бу холатларда маълум бир ички хатоликлар аникланади.

"Созлаш жараёни хатоликларни тузатиш нархи вактга тўгри пропорционал экан" -Ван Кассел

Дастурий таъминотнинг ишончлилиги сифатида маълум бир вакт жараёнида дастурий таъминот тўгри ишлаш эхтимоллигига айтилади. Дастурнинг ишончлилиги унинг ички хусусияти хисобланмайди. Дастурий махсулот яратилиш жараёнларида, иктисодий ва вактли юкотишлар асосий ўринлардан биринчида туради.

Бу юкотишларнинг сабабчиси сифатида алгоритмнинг (блоксхеманинг) сифатсиз ишлаб чикилганлигини, дастурнинг сифатсиз хужжатларга эга бўлишлигини кўрсатиш мумкин.

Статистика бўйича бир дастурчига кунига беш оператор ёзиш тўгри келади. Дастурчининг колган вакти юкорила келтирилганларни ва дастурдаги хатоликларни юкотишга кетар экан.

Компьютерный мир журнали бўйича дастурий махсулот ишлаб чикариш куввати сифатида дасьтурининг самарадорлигини эмас балки, куйидаги тушунчаларни кўзда тутамиз:

1) Дастурнинг тўгрилигини, яъни берилган масалани ечиш учун мўлжалланганлик кўрсаткичи.
2) Ишончлилик кўрсаткичи
3) Енгил ўкилувчанлиги
4) Компьютернинг хамма ресурсларидан фойдаланиш

Дастур самарадорлиги бу янги дастурнинг ярим ёки бир байт хотирани тежаб колиши ёки хамма ресурслардан умумий фойдаланиши.

Дастурий таъминотни ишлаб чикариш технологияси сифатида куйидагилар назарда тутилади:

1) ЭХМнинг барча дастурлаш воситаларини фойдаланган холда дастурлаш усуллари, дастурий таъминотнинг ишончлилиги, программавий таъминот ишини бахолашни лойихалаш сифатини ва дастурлар ишлаб чикариш воситаларини кўзда тутиб программавий таъминотни яратиш
2) Дастурий таъминот ишлаб чикариш техналогияси сифатида умумлаштирилган ва системалаштирилган бўлимлар мажмуасига дастурлаш жараёнини оптимал равишда олиб бориш йўлларига айтилади.
3) Бу жараёнда дастурий махсулот яратилишига бўлган талабдан бошлаб то махсулот фойдаланувчига топширишга ва зарур булган холда ва махсулотни узгартиришга кетган вакт киради.

Дастурлаш воситаси одатда мураккаб структуранинг барча хусусиятларини ўз ичига олади.

1) Мураккаб структуралар одатда катта сондаги модуллардан ташкил топган бўлади. Бу модуллар бирор бир умумий масалани ечишга мўлжаллаган бўлиб бир-бирлари билан ўзаро боглик бўлади.
2) Хар бир модул ўз навбатида бирор-бир кичикрок масалани ечишга мўлжалланган бўлади.

Дастурлаш воситасини яратиш давомида ЭХМнинг асосий ресурсларидан (хотира ва тезкорлик)дан самари фойдаланиш зарур. Дастурлаш воситаси учун зарур бўлган хужжатлар: Е С П Д

Бу системадаги талабларга кўра хар бир программавий махсулотни тавсифи, алгоритми ва техник хужжатлари, ички ва ташки спецификациялари бўлиши шарт.

Маълумотларни кайта ишлаш жараёнининг ахборот мухити тушунчаси. Дастур-жараённи формал тавсифлаш сифатида. Дастур воситалари хакида тушунча. Дастур воситасидаги хато тушунчаси. Тугри дастур ноконструктивлиги. Дастур воситасининг ишончлилиги. Дастурлаш технологияси, ишончли дастур воситаларини ишлаб чикиш технологияси сифатида. Жамиятни дастурлаш ва ахборотлаштириш технологияси.

Дастурлаш- маълумотларни кайта ишлаш жараёнини формал ёритиш сифатида. Дастур воситалари.

Дастурлашдан максад-маълумотларни кайта ишлаш жараёнини ёритишдир. (бундан буён факат жараён деб юритилади). ИФИПга кура маълумотларкандайдир жараёнда узатиш ва кайта ишлаш учун ярокли булган, факат ва идеяларни формаллашган турда тасвирлашдир.

Ахборот (information) эса-уларни тасвирлашда уларга бериладиган маъно. Маълумотларни кайта ишлаш (data)-маълумотлар устида бажариладиган амаллар кетма-кетлигининг систематик бажарилиши. Маълумотлар ташувчиларда сакланади.Бу ташувчилар-маълумот ташувчилари деб юритилади. Бирор кайта ишлашда фойдаланиладиган маълумот ташувчилари тупламини ахборот мухит деб атаймиз (data medium) кайсидир вакт оралигида ахборот мухитга тегишли булган маълумотлар тупламини шу ахборот мухитнинг холати деб аталади. Бир-биридан кейин келувчи, кандайдир ахборот мухитнинг холатлар кетма-кетлиги, бу-жараёндир.

Жараённи таърифлаш-берилган ахборот мухитнинг холатлар кетма-кетлигини аниклаш демакдир.

Агар биз талаб килинаётган жараённинг, кандайдир компютерда автоматик равишда амалга оширишни истасак бу таъриф формаллашган булиши керак. Бундай таъриф дастур деб айтилади. Бошка томондан, дастур инсонга тушунарли килиб тузилиши керак, чунки у дастурни ишлаб чикишда ва ундан фойдаланишда айнан кайси жараёнга дастур олиб келишини

инсон аниклаши керак. Шунинг учун, дастур инсонга кулай булган дастурлаш тилида тузилади ва шу тилдан у мой компьютер тилига, бошка, транслятор деб аталувчи дастур ёрдамида автоматик равишда утказилади. Инсон (дастурчи), узига кулай дастурлаш тилида дастурларни тузиш учун, аввало катта тайёргарлик ишларини олиб бориши керак: масала шартини аниклаш, уни ечиш усулини танлаш, талаб килинаётган дастурни куллаш спецификасини аниклаш, тузилаётган дастурини умумий ташкиллаштиришни ёритиш ва х.к. Бу ахборотдан фойдаланиш, дастурни инсон томонидан тушуниш масаласини анча соддалаштиради, шунинг учун уни алохида хужжат сифатида белгилаб куйиш максадга мувофикдир. (купинча формал булмаган, факат инсон томонидан тушуниши учун мулжалланган). Одатда дастурлар, дастур яратишда иштирок этмаган инсонлар учун яратилади. (уларни фойдаланувчилар деб аташади). Дастур фойдаланувчига тушунарли булиши учун, унинг матнидан ташкари аникланган кушимча хужжатлар талаб килинади. Дастур хужжатлари билан таъминланган, маълумотлар ташувчиларидаги дастур ёки мантикий богланган дастурлар туплами-дастур воситалари дейилади. Дастур компьютерда, малумотларни автоматик кайта ишлашни бажаради. Дастурлаш хужжатлари, ДВнинг, ихтиёрий дастури, кайси функцияни бажаришини, бошлангич маълумотларни кандай тайёрлаш керак эканлигини ва талаб килинаётган дастурни ишлатиш жараёнига узатиш, шунингдек натижани олиш нималигини тушунишга имкон беради (ёки шу дастурнинг бажарилиш самараси кандайлигини). Бундан ташкари, дастурлаш хужжатлари, шу дастурнинг узини тушунишга ёрдам беради,бу эсахусусан, уни модификациялашда зарурдир.

Тугри дастур тушунчасининг ноконструктивлиги

Шундай килиб, дастурлаш технологиясининг махсули сифатида талаб килинаётган функцияларни бажарувчи дастурларни уз ичига олувчи ДВларини тушуниш керак. Бу ерда «дастур» деганда, тугри дастур, яъни хатоси булмаган дастур тушунилади. Аммо, дастурда хато тушунчаси, дастурчилар орасида турлича талкин килинади. Майерс буйича агар дастур фойдаланувчи кутган натижани бера олмаса,унда хато бор дейилади Фойдаланувчи кутиши мумкин булган натижа, шу дастурни куллаш буйича хужжатлар асосида тузилади. Бундан келиб чикадики, дастурдаги хато тушунчаси формал эмас экан. ДВда дастурлар ва хужжатлар узаро богланган булиб, бир бутунликни ташкил киладилар. Шунинг учун дастурдаги хато тугрисида эмас, умуман, ДВсидаги хато тугрисида фикр юритилади: дастурий воситада хато бор (softwore error) деб хисобланади, агар у, фойдаланувчи кутган талабни бажара олмаса.

Хусусан, ДВсидаги хатоларнинг бир тури, бу ДВ дастурлари ва уларни куллаш буйича хужжатларнинг бир-бирига мос тушмаслигидир ишда ДВдаги хатонинг хусусий тури, яъни дастур узининг функционал спецификациясига мос келмаслиги алохида курсатилади. (бевосита дастурлашдан олдин келувчи боскичда ишлаб чикилган таърифга).

Курсатилган ишдаги бундай хато тури дастур дефекти дейилади. Аммо хатонинг бундай турини, алощида тушунча сифатида ажратиш, максадга мувофик эмас, чунки хатонинг сабаби дастурда эмас, унинг функционал хусусиятидадир. ДВсидаги топширик формал тузилмайди, чунки ДВдаги хато тушунчаси формаллашмагандир, унда ДВ тугрилигини формал усул (математик) оркали исботлаш мумкин эмас. ДВси тугрилигини, тестлаш оркали хам исботлаш мумкин эмас: Дейкстра кусатганидай, тестлаш факат ДВсида хато борлигини курсатади. Шунинг учун тугри ДВси тушунчаси шу маънода, яъни ДВсини яратиш иши охирлаганидан кейин, биз максадга эришганимизни била олишимиз маъносида конструктив эмас.

Дастур воситаларининг ишончлилиги. Тугри ДВсининг алтернативи, бу, ишончли ДВсидир. ДВсининг ишончлилиги-берилган шароитларда ва берилган вакт оралигида аник вазифаларни катта эхтимол билан, инкор этмасдан бажариш кобилиятига айтилади. Бунда ДВсида инкор этиш тушунчаси сифатида, унда хато борлиги тушунилади . Шундай килиб, ишончли ДВсида хам хатолар булиши мумкин-факат бу хатолар, шу ДВсини, берилган шароитларда, амалий куллашда етарли даражада кам учраса булди. ДВси шундай хусусиятга эга эканлигини, уни тестлаш йули билан ва уни амалиётда синаб аникланади. Шундай килиб биз, аслида, тугри ДВлари эмас, факат ишончли булган ДВларини ишлаб чикамиз.

ДВсининг ишончлилиги турли даражада былади. Бу даражани кандай улчаш мумкин? Техникадагидай, ишончлилик даражасини ДВси аник вакт оралигида инкор этмай ишлаш эхтимоли оркали характерласак булади. Аммо, ДВсининг узига хос хусусиятларига кура, бу эхтимолни аниклаш, шу масалани техникада ечишга караганда анча кийинрокдир. Кейинрок бу масалани муфассалрок мухокама киламиз. ДВсининг ишончлилик даражасини бахолашда, хар

бир инкор этишнинг натижаларини эътиборга олишимиз керак. ДВдаги баъзи хатолар, уни кулланилишида, факат баъзи нокулайликларни келтириб чикарса, баъзи, бошка хатолар дахшатли натижаларни келтириб чикариши мумкин, масалан, инсон хаётига хавф солиши мумкин. Шунинг учун, ДВсининг ишончлилигини бахолаш учун, фойдаланувчи учун хар бир инкор этишнинг, канчага тушишини эътиборга олувчи, кушимча курсаткичлардан фойдаланилади.

Дастурлаш технологияси-ишончли дастур воситаларини ишлаб чикиш технологияси сифатида.

«Технология» сузининг одатдаги маъносига боглик равишда (1.6) дастурлаш технологияси (programming technology) деганда, талаб килинаётган ДВларини яратишга олиб келадиган ишлаб чикиш жараёнларининг тупламини ва шунингдек, шу жараёнлар тупламини ёитишни тушунамиз. Бошкача айтганда, дастурлаш технологияси тушунчасини, биз, кенг маънода, дастур воситаларини ишлаб чикиш технологияси маъносида тушунамиз. Бунга шу воситани яратиш фикри мавжуд булган вактдан бошлаб барча жараёнлар ва хусусан, зарур дастур хужжатларини яратиш билан боглик жараёнлар хам киради. Бу тупламнинг хар бир жараёни кандайдир усул ва воситалардан фойдаланишга асосланади, масалан, компьютер (бу холда биз дастурлашнинг компьютерли технологияси тугрисида фикр юритамиз). Адабиётларда, дастурлаш технологиясига, бир-биридан фаркли таърифлар мавжуд. Бу таърифлар ишда мухокама килинади. Адабиётларда, дастурлаш технологиясига якин булган, дастурлаш инженерияси тушунчаси хам булади. Ишланмага, ундан фойдаланишга,уни кузатиб боришга ва дастур воситаларининг муомаладан олиб ташлашга системали ёндашув сифатида аникланувчи дастур инженерияси тушунчаси хам адабиётларда кулланилади. Юкорида эслатиб утилган иш айнан дастур инженериясига багишланган. Дастурлаш технологияси ва дастурлаш инженерияси орасидаги аросий фарк бу материални караб чикиш усули ва системалаштиришдадир . Дастурлаш технологиясида эътибор, ДВни ишлаб чикиш жараёнларини урганишга (технологик жараёнларни) ва уларнинг бажарилиш тартибига берилади- бу жараёнларда ДВси ишлаб чикишнинг усуллари ва инструментал воситалари, фойдаланилади. (уларни куллаш технологик жараёнларни ташкил килади). Дастур инженериясида эса, ДВси ишлаб чикишнинг турли усул ва инструментал воситалари, аник максадларга эришиш нуктаи-назаридан урганилади-бу усул ва воситалар турли технологик жараёнларда фойдаланилади (ва турли дастурлаш технологияларида хам).

Дастурлаш технологиясини дастурлаш услубияти деб тушуниш керак эмас . Дастурлаш технологиясида услублар «юкоридан» -технологик жараёнларни ташкил килиш нуктаи-назаридан каралади, дастурлаш услубиятида, эса, услублар «пастдан», уни куриш асослари нуктаи-назаридан каралади бет ишда дастурлаш услубияти, дастурли таъминлашни ишлаб чикиш жараёнида кулланиладиган ва битта умумий фалсафий ёндошувга бирлаштирилган механизмлар туплами сифатида каралади.)

Ишончлилик, ДВсининг ажралмас хусусияти булгани учун, биз дастурлаш технологиясини, ишончли ДВни ишлаб чикиш технологияси деб караймиз. Бу куйидагиларни билдиради:

-биз, ДВни ишлаб чикиш жараёнларини, ДВ гояси пайдо булган вактдан бошлаб караймиз.

-бизни, дастурлаш конструкцияларини куриш масалалари кизиктирибгина колмай, одам идрок килиш(ноформал) нуктаи-назаридан функцияни ва кабул килинадиган ечимларни еритиш масалалари хам кизиктиради.

-технология махсулоти сифатида ишончли(лекин, хар доим хам тугри булавермайдиган)ДВси кабул килинади.

Дастурлаш технологиясига бундай нуктаи-назардан караш, технологик жараёнларнинг ташкил килинишига, ундаги усул ва инструментал воситаларни танлашга жиддий таъсир курсатади.

Дастурлаш технологияси ва жамиятни ахборотлаштириш.

Дастурлаш технологияси, дастурлаш ривожининг хар хил этапларида турлича рол уйнади. Компьютерлар куввати ва воситалар ривожланиши ортиши ва дастурлаш услуби ривожланиши билан, компьютерда ечиладиган масалалар мураккаблиги хам орта бошлайди, бу эса дастурлаш технологиясига юкори эътибор каратилишига сабаб булади. Компьютерлар ва

айникса, компьютер ташувчиларида ахборотни саклаш нархининг бирдан тушиб кетиши, компьютерларни инсон фаолиятининг барча доираларида кулланилишига олиб келди ва бу эса уз навбатида дастурлаш технологиясининг йуналишини жиддий узгартирди. Инсон фактори унда хал килувчи рол уйнай бошлади. ДВси сифатининг етарли даражада кенг тушунчаси тузила бошлади, шу билан бирга, унинг афзаллиги сифатида, ДВсининг самарадорлиги эмас, фойдаланувчининг у билан ишлаш кулайлиги тушунила бошлади. (ишончлигини эътиборга олмай туриб). Компьютер тармокларининг кенг кулланилиши, таксимланган хисоблашларнинг, ахборотга дистанцион киришнинг ва одамлар орасидаги маълумотлар алмашишнинг электрон усулининг интенсив ривожланишига олиб келди. Компьютер техникаси, айрим масалаларни ечиш воситаси булибгина колмай, оддийгина, инсонни кизиктирадиган саволларга жавоб беради, реал ва абстракт дунёни ахборот моделлаштириш воситасига айланиб бормокда.Инсонлар жамиятини, чукур ва тула ахборотлаштириш (компьютерлаштириш) боскичи бошланмокда. Буларнинг хаммаси, дастурлаш технологияси олдида, янги ва етарли даражада кийин муаммоларни тугдирмокда.Бир неча 10 йил ичида дастурлашнинг ривожланишига кискача таъриф берамиз. 50-йилларда компьютерлар куввати (биринчи авлод) катта эмас эди, улар учун, дастурлаш, машина кодида бажарилар эди. Асосан, илмий-техник масалалар (формулалар буйича хисоблаш), дастурлашдаги топширик, коидага мувофик, масаланинг етарли даражада аник куйилишини уз ичига олади. Интуитив дастурлаш технологиясидан фойдаланилди: топширик буйича дарров дастур тузишга киришилар эди ва бунда топширик бир неча марта узгартирилар эди(бу эса,итерацион жараён булган дастур тузиш вактини узайтиради) Энг кам хужжатлаштириш, дастурлаш бошланганидан сунг расмийлаштирилар эди. Шунга карамай, айнан шу даврда, машинали кодда дастурлашдаги кийинчиликларни бартараф килишга мулжалланган, дастурлаш технологияси учун, фундаментал булган модулли дастурлаш концепцияси вужудга келди. Биринчи юкори даражадаги дастурлаш тиллари юзага келди ва улардан, ФОРТРАН кейинги ун йилликларда фойдаланиш учун енгиб чикди.

60-йилларда юкори даражали (АЛГОЛ 60, ФОРТРАН, КОБОЛ, ва бошкалар) дастурлаш тилларининг, шиддат билан ривожланиши ва кенг кулланилишини кузатамиз. Бу тилларнинг дастурлаш технологиясида тутган урни кузга куринарли булиб, колди. Бу тиллар, катта дастурларни ишлаб чикиш жараёнида, вужудга келадиган барча муаммоларни ечади, деган ишонч, узини окламади. Компьютер кувват ошиши ва юкори даражадаги тилларда дастурлаш тажрибасининг ортиши билан компьютерда ечилган масалаларнинг мураккаблиги хам ортар эди, бунинг натижасида, дастурни модулли ташкил килиш эътиборга олинмаган тилларнинг чегараланганлиги билиниб колди. Факат, модулли дастурлаш имкониятини саклаган ФОРТРАНгина, кейинги, ун йилликларда олдинда борди. Бундан ташкари, биз кайси тилда дастурлашимиз эмас, кандай дастурлашимиз ахамиятга эга булиб колди . Бу дастурлаш технологияси ва услубияти устида жиддий фикр юритишнинг бошланиши сабаб булди. 2-давр компьютерларидаги узилишнинг пайдо булиши мультидастурлаш ва катта дастурлаш системаларини яратилишига олиб келди. Ишланмаларни коллектив булиб бажариш, катор жиддий техник муаммоларни юзага келтирди .

70-йилларда, ахборот мухитлар ва маълумотлар базаси кенг таркала бошлади. 70-йиллар урталарида компьютер ташувчиларида 1 битли ахборотни саклаш нархи, анъанавий ташувчиларга караганда анча кам булиб колди. Бу эса маълумотларни саклаш компьютерли системаларига кизикиш орттирди. Дастурлаш технологиясининг интенсив ривожланиши энг аввал куйидаги йуналишларда бошланди.

-пасаювчи ишлаб чикиш ва таркибли дастурлашни асослаш ва кенг тадбик килиш.

-маълумотларнинг абстракт типини ва модулли дастурлаш ривожланиши (хусусан, хусусиятлари ва модулларни амалга ошириш буйича булиниш гоясининг тугилиши ва маълумотлар таркибини яширувчи модуллардан фойдаланиш).

-ДВ мобиллиги ва ишончлилигини таъминлаш муаммоларини урганиш.

-ДВини коллектив ишлаб чикишни бошкариш услубини яратиш.

-дастурлаш технологиясини куллаш учун инструментал дастурлаш воситаларининг пайдо булиши.

80-йиллар, персонал компютерларнинг инсонлар фаолиятининг барча доирасида кенг кулланилиши билан ва ДВ фойдаланувчиларининг кенг ва турли-туман таркиби купайиши билан характерланади. Бу эса фойдаланиш интерфейсларининг ва ДВси сифатининг аник

концепциясини яратишнинг тез ривожланишига сабаб булади. Дастурлаш технологияси талабларини эътиборга олувчи дастурлаш тиллари пайдо була бошлади (масалан, АДА). ДВси тиллари хусусиятлари ва усуллари ривожлана бошлади. Технологик жараёнлар энг аввал, бу жараёнларда яратиладиган хужжатларни тез суръатда стандартлаштириш бошланади. ДВсини ишлаб чикишга объектли ёндашув олдинги позицияларга чикади. Турли инструментал ишлаб чикиш ва кузатиб бориш мухитлари яратилади Компьютер тармоклари концепцияси ривожлана бошлади.90-йилларда барча инсонлар жамиятида, халкаро компьютер тармоги кенг таркала бошлади, унга персонал компютер терминал сифатида улана бошланди. Бу эса навбатдаги компьютер тармоги ахборотига киришни тартибга солишда катор муаммолар тугдирди. Компьютердаги ахборот ва тармокардаги маълумотларни химоялаш муаммоси тугилди. ДВ ишлаб чикиш компьютерли технологияси (CASE-технология) ва у билан боглик дастурни спецификациялашнинг формал методлари тез ривожлана бошлади. Жамиятни тулик ахборотлаштириш ва компьютерлаш янги боскичи бошланди.

Назорат савлоллари

1. Дастурлаш фанининг тарихий боскичлари.
2. ДТ ишончлилиги.
3. Дастурлаш- маълумотларни кайта ишлаш жараёнини формал ёритиш сифатида.
4. Тугри дастур тушунчасининг ноконструктивлиги

Фойдаланилган адабиётлар

6. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. –СПб: Питер, 2003.- 396 с.
7. Афанасьев А.Н. Формальные языки и грамматики: Учебная школа: УлГТУ, 1997. – 84 с
8. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции -: Мир, 1979.- 487с.
9. Компаниец Р.И. Системное программирование. Основы построения трансляторов. СПб.:Корна принт., 2000. -256 стр.
10. Дьяконов В.Ю. Системное программирование. Высш.шк.. 1990. -221 с.

Маъруза №4. Мавзу : Дастурлаш тизимлари таркиби.

Режа:

1. **Дастурлаш тизимлари**
2. **Идентификаторлар жадвали**

Дастурлаш тизимлари тил муаммоларини хал килувчи дастурларни бирлаштирадилар ва дастурий таъминотни ишлаб чикаришга мулжалланган.

Дастурлаш тизимларига 1) трансляторлар; 2) кутубхона дастурлари; 3) редакторлар; 4) компановщиклар; 5) загрузчиклар;6) отладчиклар кирадилар.

Дастурларга хизмат курсатувчи тизимлар – бу махсус сервис дастурлар булиб, улар ёрдамида операцион тизимни узига хизмат курсатиш мумкин.

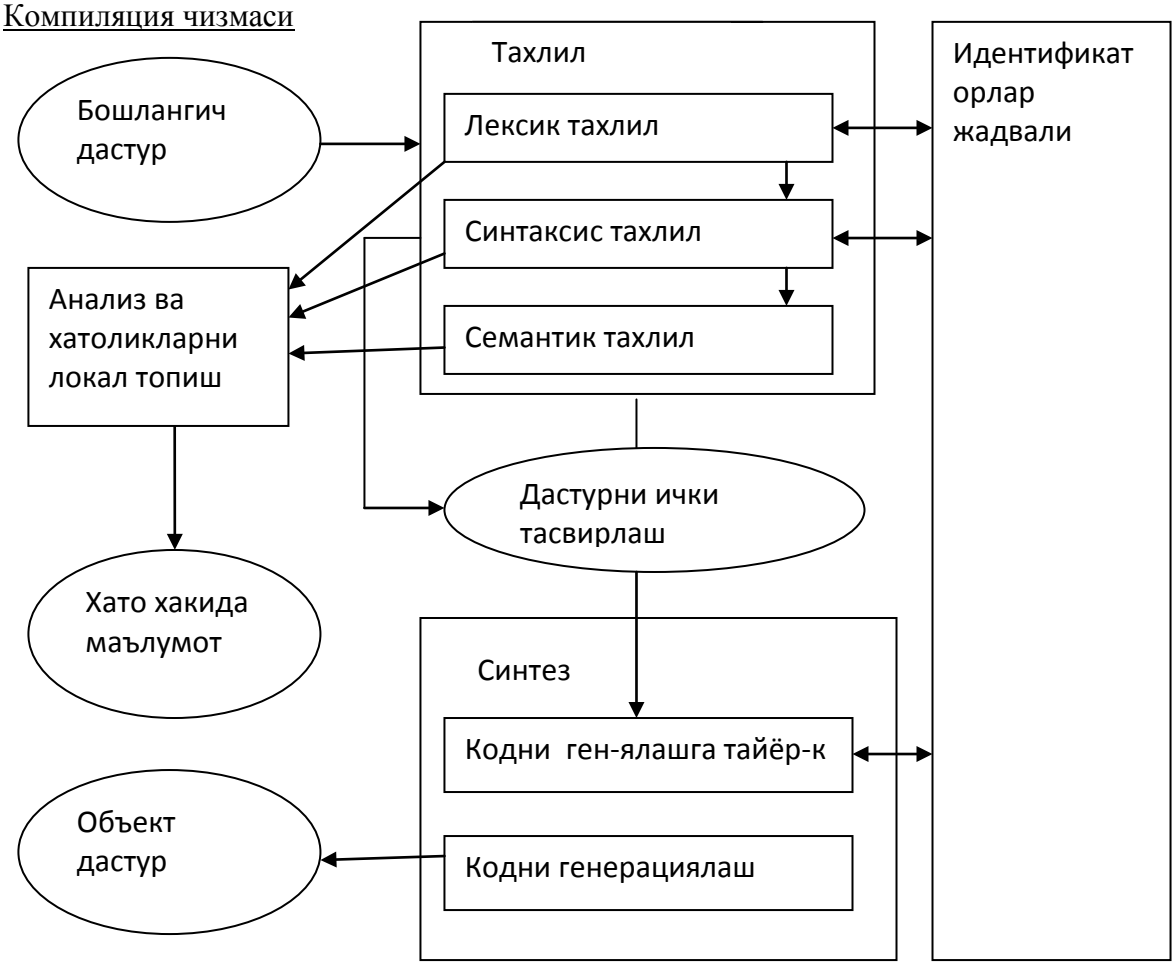
Транслятор – бу дастур берилган дастурлаш тилидаги кирувчи дастур матнини унга эквивалент булган чикишдаги натижавий тилга угиради.

Компилятор – бу транслятор булиб, у берилган дастур мантнини унга эквивалент булган машина командаларидаги объект дастурга угиради.

Интерпретатор – бу дастур булиб, у берилган дастур матнини бирданига кабул килади ва бажаради (натижавий коди булмайди)

Компилятор формал тиллар нуктаи назаридан куйидаги 2 асосий функцияларни бажаради:1) у кирувчи дастур матни тили учун англовчи хисобланади (кирувчи дастур занжирлар генератори булиб хисобланади);2)натижавий дастур тили учун генератор хисобланади (англовчи булиб хисоблаш тизими хисобланади)

Компиляция чизмаси



Лексик тахлил – бу компилятор булаги булиб, дастур литераларини укийди ва улар оркали кирувчи тил лексемаларини куради.

Синтаксис тахлил – Тахлил боскичидаги компиляторнинг асосий булагидир. Тилнинг систаксис конструкцияларини ажратади.

Семантик тахлил – бу компилятор булаги булиб, кирувчи тил семантикаси нуктаи назаридан дастур матнини текширади.

Кодни генерациялашга тайёргарлик – натижавий дастурнинг синтези билан боглик булган харакатларга тайёргарлик бажарилади.

Кодни генарациялаш – натижавий кодни бевосита хосил этиш – кодни оптимизациялашни уз ичига олган асосий фаза.

Идентификаторлар жадвали – кирувчи дастур элементлари хакидаги маълумотларни сакловчи берилганлар туплами. Бир неча хил идентификаторлар жадвали мавжуд.

Утиш – бу ташки хотирадан берилганларни охирги укиш жараёни, уларни кайта ишлаш ва ташки хотирага жойлаштириш. Компиляциянинг бир фазаси - бир утишдир.

2.Идентификаторлар жадвали

Идентификаторлар жадвали – кирувчи дастур элементлари хакидаги маълумотларни сакловчи берилганлар туплами. Бир неча хил идентификаторлар жадвали мавжуд.

ИЖ ташкил этиш: Идентификаторга боглик холда тупламда турлича маълумотлар сакланади. 1)Узгарувчи-исм,тур,адрес 2)Константа-исм, агар бор булса,адрес,киймат;3)Функция аргументлар сони,тур,исм,адрес

ИЖ лексик тахлил фазасида ташкил этилади ва кетма-кет тулдирилади. Лексик тахлил фазасида идентификаторлар турлари ёзилади. Кодни генерациялашга тайёргарлик фазасида – хотира ажратилади.

ИЖ ини ташкил этишнинг асосий критерийси булиб кидирув вакти хисобланади, чунки асосий функциялар ёзувлврни кидирув ва кушишдан иборат (купрок кидирув амалга оширилади)

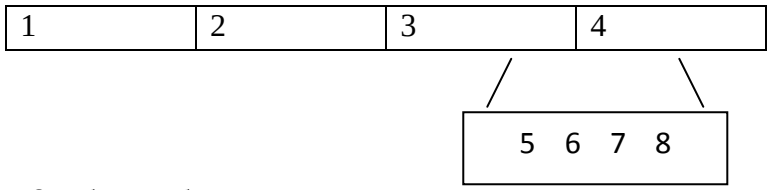
I. Тартибланмаган руйхат – ИЖ да элементлар келиб тушиши буйича кушидали.

II. Тартибли руйхат – Иждаги барча элементлар табиий тартибга мос равишда ошиб бориш ва камайиш буйича тартибланган.

III. ИЖ ининг бинар дарахти бинар дарахт куринишига эгадир. Хар бир элемент ёки чукки - идентификатордир. Дарахтнинг илдиз чуккиси булиб, биринчи келиб тушган идентификатор хисобланади.

Бинар дарахтнинг формаси келиб тушаётган идентификаторларга боглик. Камчилиги дарахтдан тартибли руйхатни келитириб чикариш мумкин.

Идентификаторлар жадвалига барча калит суз хисобланмаган исмлар киритилади. У куйидаги куринишга эга:



3,4,5,6,7,8 – бошка боскичларда тулдирилади;

1 – тартиб раками;

2 – исм узунлиги байтларда;

3 – хотирадиги исм белгиларини саклаш манзили;

4 – идентификатор атрибутлари;

Атрибутлар сифатида:

5 – куриниш (ёки оддий узгарувчи исми, ёки туплам, ёки процедура (функция), метка);

6 – тур (хакикий, бутун, каторли, белгили, мантикий);

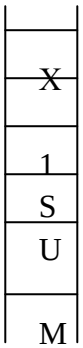
7 – хотира синфи (статик, автоматик, динамик, ташки);

8 – улчови (масалан, катор учун – узунлик, туплам учун – индекслар сони, интервьюлашган турлар учун – чап ва унг чегаралар киймати);

Маълумотларни исмларда тасвирлашнинг шундай вариант мавжудки унда барча исмни ташкил этувчи символлар занжирни ташкил этадилар ва хар бир янги исм Ушбу занжирнинг охирига уланадилар.

Исмлар жадвали

Ракам	Исм узунлиги	Бошлангич манзил
1	2	
2	3	



Var X1, sum: integer

Чикувчи жадвал – лексемалар кодлари жадвали (синтаксис тахлил учун бошлангич).

Жадвал структураси:

Лексема раками (дастурда)	Тури	Лексема раками (жадвалда)
1		
2		

Биринчи майдонда – дастурдаги лексемани пайдо булиш раками.

Учинчи майдонда – мос жадвалдаги ракам. Агар исм бир неча марта кайтарилса, у холда лексемалар кодлари жадвалида унга шунча марта катор мос келади.

Иккинчи майдонга исм хакидаги маълумот киритилади: исм – I, терминал белгилар – T, константалар – C.

Жадвални ташкил этишга мисол караймиз.

```
Program authm;
Var
        I, J, Sum: Integer;
Begin
        Sum:=0;
        For i:=1 to 100 do;
        Begin
                Read(J);
                Sum:=Sum+J;
        End
        Sum:=Sum div 100;
        Write(Sum);
End
```

Терминал белгилар жадвали

№	Белги	Тури		
		Ажратувчи	Амал ишораси	Калит Суз
1	**;**	1	0	0
2	**,**	1	0	0
3	**:**	1	0	0
4	пробел	1	0	0
5	**:=**	0	1	0
6	**(**	1	0	0
7	**)**	1	0	0

8	+	0	1	0
9	Div	0	1	0
10	Program	0	0	1
11	Var	0	0	1
12	Integer	0	0	1
13	Begin	0	0	1
14	For	0	0	1
15	To	0	0	1
16	Do	0	0	1
17	End	0	0	1
18	Read	0	0	1
19	Write	0	0	1
20	.	1	0	0
21	-	0	1	0
22	*	0	1	0

Исмлар жадвали

№	Исм
1	Arithm
2	I
3	J
4	Sum

Константалар жадвали

№	Константа	Асос0	Тури	Аниклик
1	0	10	Бутун	2
2	1	10	Бутун	2
3	100	10	Бутун	2

Лексемалар коди

№	Тури	№ жадвалдаги раками	Лексемалар
1	T	10	Program
2	T	4	Пробел
3	I	1	Arithm
4	T	1	;
5	T	11	Var
6	T	4	Пробел
7	I	2	I
8	T	2	,
9	I	3	J
10	T	1	;
11	…	…	…

Лексема кодларининг чикувчи жадвалида пробел хакидаги маълумот жойлашмайди.

Назорат саволлари

1.Қуйи даража дастурлаш тиллри?

2.Юқори даража дастурлаш тиллари?

3.Компльятор ва интепритатор фарқи?

Фойдаланилган адабиётлар

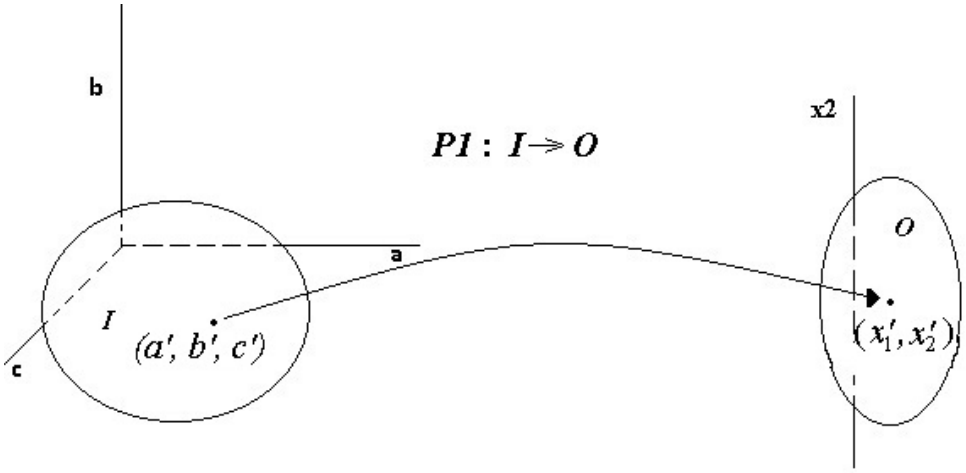
11. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. –СПб: Питер, 2003.-396 с.
12. Афанасьев А.Н. Формальные языки и грамматики: Учебная школа: УлГТУ, 1997. – 84 с
13. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции -: Мир, 1979.-487с.
14. Компаниец Р.И. Системное программирование. Основы построения трансляторов. СПб.:Корна принт., 2000. -256 стр.
15. Дьяконов В.Ю. Системное программирование. Высш.шк.. 1990. -221 с.

Маъруза №5. Мавзу: Транслятор, компиляторва интерпретатор тушунчалари

Режа:

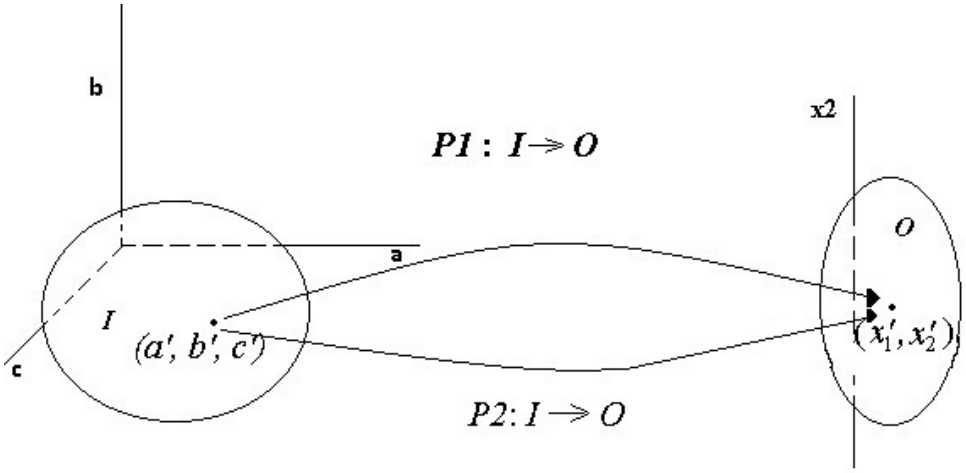
1. **Транслятор**
2. **Компилятор**
3. **Интерпретатор**
4.

Транслятор *деб бошланғич тилдаги дастурни натижавий тилдаги эквивалент дастурга ўгирувчи дастур тушунилади.* Эътибор қилган бўлсангиз ушбу таърифда уч марта дастур сўзи ишлатилмоқда. Биринчи навбатда транслятор - бу компьютер дастури. Бошқа хар қандай дастур каби транслятор хам бошланғич маълумотларни қайта ишлаб натижа ҳосил қилади. Транслятор учун бошланғич маълумот - бошланғич тилдаги дастур ҳисобланади. Транслятор ишининг натижаси эса натижавий тилдаги дастур. Масалан Borland корпорацияси томонидан ишлаб чиқилган C++ тилининг трансляторини қарайлик. Сиз ёзган prog1.cpp номли файл ушбу транслятор учун бошланғич маълумот бўлиб ҳисобланади. Транслятор ишини тугатгандан сўнг, трансляция натижаси - prog1.exe файли ҳосил бўлади. Ушбу файл Сизнинг C++ ёзган дастурингизнинг машина тилидаги кўриниши бўлиб ҳисобланади. Шу ўринда эквивалент дастур тушунчасини аниқлаштириб ўтиш лозим. Фараз қилайлик P_1 бирор дастур бўлсин, унинг қайси тилда ёзилганлигининг аҳамияти йўқ. Биз ушбу дастурни бошланғич маълумотлар тўплами I ни натижавий маълумотлар тўплами O га акслантириш сифатида талқин қилишимиз мумкин (расмга каранг).



Расм 1 Дастур акслантириш сифатида

Ушбу расмда $ax^2 + bx + c = 0$ квадрат тенгламани ечиш учун мўлжалланган P_1 дастур учун бошланғич қийматлар тўплами I ва натижавий қийматлар тўплами O кўрсатилган. Фараз қилайлик, P_2 – квадрат тенгламани ечиш учун мўлжалланган дастур бўлсин. Ҳар иккала дастур ихтиёрий бир ҳил бошланғич қийматлар учлиги (a', b', c') учун бир хил натижа (x_1', x_2') қайтарса бу дастурлар эквивалент дейилади. Аниқроғи P_1 ва P_2 дастурлар эквивалент дастурлар дейилади, агар $\forall (a', b', c') \in I$ учун $P_1(a', b', c') = P_2(a', b', c') = (x_1', x_2') \in O$ шарт бажарилса. Эквивалентлик тушунчасини изоҳловчи расм қуйида келтирилган (расм 2 га қаранг). Трансляция жараёнига нисбатан эквивалентлик тушунчаси қуйидагини англатади. Алгоритмик тилдаги ҳар қандай P1 дастур маълумотларни қайта ишлашнинг бирор жараёнини тавсифидан иборат, трансляция натижасида ҳосил қилинган натижавий тилдаги P2 дастур ҳам айнан шу жараённи тавсифлайди.



Расм 2. Дастурларнинг эквивалентлигини изоҳловчи расм

Трансляция жараёни икки босқич: таҳлил (анализ) босқичи ва натижавий тилдаги дастурни ҳосил қилиш (синтез) босқичидан иборат. Анализ босқичида бошланғич тилдаги дастур таҳлил қилинади ва дастурнинг ички кўриниши ҳосил қилинади. Синтез босқичида ушбу ички кўриниш асосида натижавий тилдаги дастур ҳосил қилинади. Трансляция жараёнини биринчи яқинлашишда қуйидагича тасаввур қилиш мумкин.

Расм 3. Трансляция жараёнининг соддалаштирилган схемаси.

Шундай қилиб, транслятор иккита вазифани бажаради: 1) Агар бошланғич тилдаги дастурда ҳеч қандай хато бўлмаса, у натижавий тилдаги дастурни ҳосил қилади; 2) аксинча бошланғич тилдаги дастурда хатолар бўлса, у ҳолда хатолар учраган сатр ва хатолик типи ҳақида маълумот хабар беради.

Компилятор деб бошланғич тилдаги дастурни машина кодларига ўгирувчи транслятор тушунилади. Компилятор томонидан ҳосил қилинган дастур **объектли код** деб аталади. Аксарият ҳолларда объектли код бевосита машинада бажариш учун яроқли бўлмайди. Бунинг сабаби шундаки, биринчидан барча дастурлаш тилларининг компиляторлари, амалиётда кўп учрайдиган дастурлаш масалаларини ҳал қилиш учун мўлжалланган қисм дастурлар тўплами (қисм дастурларнинг тизимли кутубхонаси) билан бирга тақдим қилинади. Ушбу қисм дастурлар ўз зиммасига киритиш-чиқаришни ташкил этиш, математик ва бошқа кенг тарқалган функцияларни амалга оширишни олади. Иккинчидан, фойдаланувчининг ўзи ҳам шахсий ёки бошқа қисм дастурлар тўпламига эга бўлиши ёки ишлаб чиқилаётган дастурий восита алоҳида компиляция қилинадиган бир неча дастурий файллардан ташкил топиши мумкин. Шу сабабли агар дастурда тизимли кутубхонага тегишли қисм дастурга ёки бошқа дастурий файлдаги қисм дастурга мурожаат бўлса, бундай мурожаатлар объектли кодни ҳосил қилиш даврида тўлиқ аниқланган бўлмайди. Чунки қисм дастурни чақириш учун бу қисм дастурнинг кириш нуқтаси адресини билиш керак, қисм дастур бошқа файлда бўлганлиги, баъзи ҳолларда хаттоки ҳали мавжуд эмаслиги туфайли, объектли кодни хосил қилиш пайтида ташқи қисм дастурнинг кириш нуқтасини аниқлашнинг иложи йўқ. Бу муаммо йиғиш жараёнида ҳал этилади. Йиғиш

жараёнида бир қанча ўзаро боғланган объектли код файллари ягона бажариладиган файлга бирлаштирилади. Бу статик боғлаш дейилади. Ушбу ишни бажарувчи дастур йиғувчи (компоновщик) ёки боғловчи (линкер) дейилади. Статик боғлашдан ташқари ҳозирда динамик боглаш кенг қўлланилади. Динамик боғланувчи файлларнинг ёрқин мисоли бу Windowsнинг DLL файлларидир. Динамик боғлашнинг статик боғлашдан фарқи шундаки, бошқа дастурий файлдаги қисм дастурларнинг кириш нуқтаси асосий бажарилувчи файл хотирага юкланиб, бошқарув узатилганда аниқланади.

Интерпретатор сифатида бошланғич тилдаги дастурни таҳлил қилиб, ҳосил қилинган оралиқ кўриниш асосида (натижавий кодни ҳосил қилмасдан) ушбу дастурни бажарувчи дастур тушунилади. Бошқача айтганда интерпретаторнинг иш натижаси бошлангич дастурни бошланғич қийматлар билан бажаришдан олинган қийматлардан иборат. Юқорида айтилганлардан кўринадики, интерпретаторнинг ишлаши кўп жиҳатдан трансляторнинг иш жараёнига ўхшаб кетади, асосий фарқ транслятор ички кўринишдан натижавий тилдаги дастурни ҳосил қилса, интерпретатор ички кўриниш асосида дастурнинг бажарилишини ташкил қилади, шу билан бирга дастурнинг ички кўриниши интерпретатор ишини тугатиши билан изсиз йўқ бўлиб кетади.

Интерпретаторларнинг энг қулай жиҳати, бу дастурни сатрма-сатр киритиш чоғидаёқ бажариб кўриш имкониятидир. Шу сабабли интерпретация қилинадиган дастурлаш тилларида дастурлашни осон ўрганиш мумкин деб ҳисобланади. Шу билан бирга интерпретация қилинадиган дастур, компиляция қилиниб, машина тилига ўгирилган дастурдан бир неча баробар секин ишлайди, ундан ташқари ҳар сафар интерпретациядан олдин таҳлил бажарилади. Бу фойдаланувчи қўл остида ҳар доим интерпретатор бўлишини талаб этади. Компиляторнинг устунлиги, компиялция натижасидан (машина кодидаги бажариладиган файлдан) бир неча марталаб фойдаланиш мумкинлигидадир. Хозирги кунда замонавий дастурлаш тизимлари нафақат дастурни машина тилига компиляция қилиш, балки сатрма-сатр бажариш имконини ҳам беради. Ундан ташқари аксарият интерпретация қилинадиган тиллар учун компиляторлар ҳам мавжуд.

Маъруза №6. Мавзу: Транслятор, компиляторва интерпретатор ишлаш тамойиллари.

Компилятор белгилар каторини текшириш муаммосини, ушбу катор шу тилга тегишли ёки йуклигини аниклаш учун ва тегишли булса, у холда тугилувчи грамматика коидалари терминларида каторни структурасини англашни хал килиши керак. Ушбу муаммо разбор муаммоси сифатида машхур. Тугилувчи коидалар билан ишловчи грммматикани текширамиз. (Е-бошлангич белги).

1. E - E+T
2. E- T
3. T - T*F
4. T - F
5. F- (E)
6. F - x
7. F - y

Куриниб турибдики, (x+y)*x катор ушбу тилга тегишли. Хусусий холда, буни куйидагича келтириб чикариш мумкин (хар бир келтириб чикариш кадами учун кулланилаётган коида раками курсатилган):

2) E T
3) T*F
4) F*F
5) (E)*F
1) (E+T)*F
2) (T+T)*F
4) (F+T)*F
6) (x+T)*F
4) (x+F)*F
7) (x+y)*F
6) (x+y)*x

Ёки куйидагича келтириб чикариш мумкин:

2) E T
3) T*F
6) T*x
4) F*x
5) (E)*x
1) (E+T)*x
4) (E+F)*x
7) (E+y)*x
2) (T+y)*x
4) (F+y)*x
6) (x+y)*x

Биринчи чикишнинг хар бир боскичида сентенциал форманинг энг чап нотерминали грамматиканинг бирон бир тугилувчи коидаси ёрдамида алмаштирилди. Шу сабабли ушбу чикиш **чап томонли чикиш** дейилади. Хар бир боскичида энг унг нотерминал алмаштирилган иккинчи чикиш эса **унг томонли чикиш** деб аталади.

Шу каби бошка чикишлар хам мавжудки, улар чап томонли хам, унг томонли хам хисобланмайдилар, лекин трансляторларни куришда улардан фойдаланилмайди. **Гапни чап томонли разбори** чап томонли чикишни кулланилган холда гапни генерация килиш учун тугилувчи коидаларнинг кетма-кетлиги сифатида аникланади. Ушбу холда чап томонли разборни куйидагича ёзиш мумкин: 2,3,4,5,1,2,4,6,4,7,6.

Гапни унг томонли разбори унг томонли чикишни кулланилган холда гапни генерация килиш учун тугилувчи коидаларнинг тугилувчи коидаларнинг тескари кетма кетлиги хисобланади; масалан, юкорида келтирилган холда унг томонли разбор куйидагича ёзилади: 6,4,2,7,4,1,5,4,6,3,2.

Тугилувчи коидалар кетма-кетлигининг тескари тартиби шу билан богликки, унг томонли разбор гапни бошлангич белгига келтириш сифатида каралади, гапни бошлангич белгидан бошлаб генерация килиш эмас (пастдан юкорига караб разбор). Шуни таъкидлаш керакки, хар бир тугилувчи коидадан иккала чикишда хам бир хил сон марта фойдаланилади (разборларда).

Разбор дарахти. Чикиш яна синтаксис дарахт (разбор дарахти) номи билан машхур дарахтни куриш терминларида хам ифодаланиши мумкин. (x+y)*x катор холида синтаксис дарахт куйидаги куринишда булади:

Разбор муаммисини куйидаги масалаларга келтириш мумкин.

1. чап томонли разборни топиш
2. унг томонли разборни топиш
3. синтаксис дарахтни куриш.

Бир хил кийматга эга булмаган грамматикалар.

Купгина холларда чап томонли ва унг томонли разборлар ва синтаксис дарахт уникал хисобланади. Лекин, куйидаги тугилувчи коидали грамматика учун

S S+S| x x+x+x гап иккита синтаксис дарахтга, иккита чап (унг) томонли разборга эга.

S S+S	S S+S
S +S+S	x+S
x+S+S	x+S+S
x+x+S	x+x+S
x+x+x	x+x+x

Агар грамматикада генерация килинган кандайдир гап, биттадан ортик разбор дарахтига эга булса, бундай грамматика хакида у бир кийматли эмас дейилади. Эквивалент шарт шунда куринадики, гап биттадан ортик чап ёки унг томонли разборга эга булиши керак. Грамматиканинг бир кийматли эмаслигини урнатиш масаласи умумий холда ечими йук масаладир, яъни киришда ихтиёрий грамматикани кабул киладиган ва уни бир кийматлими

ёки йуклигини аниклайдиган универсал алгоритм мавжуд эмас. Баъзи бир бир кийматли булмаган грамматикаларни уша тилни генерация киладиган бир кийматлига айлантириш мумкин. Масалан, куйидаги тугилувчи коидаларга эга грамматика

S x S +x бир кийматли булиб, у уша тилни худди аввалги бир кийматли булмаган грамматика каби генерациялайди.

Разбор усуллари купинча пастдан юрувчидир, яъни бошлангич белгидан бошлаб гапга караб юрилади ёки юкоридан юрувчи булиб, гапдан бошлаб бошлангич белгига караб юрилади.

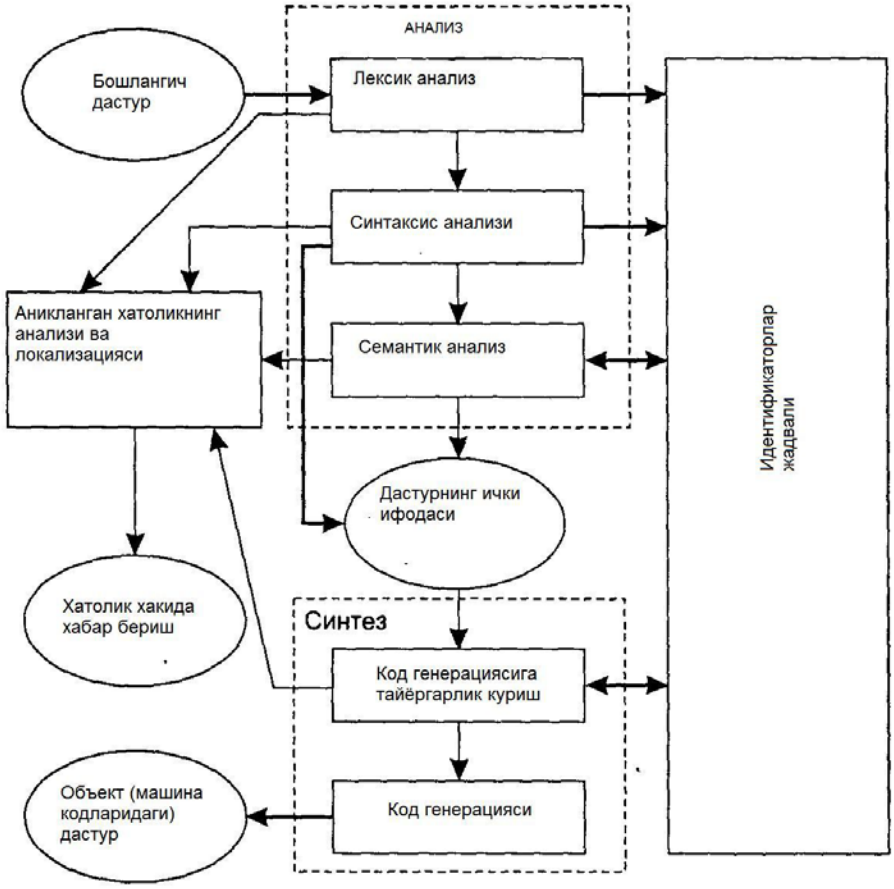
Разборда ечимни рад этиш (отказ) кайтиш (возврат) деб аталади. Разбор усуллари кайтиш бор ёки йуклигига караб детерминирован ва нодетерминирован булиши мумкин. Нодетерминирован усуллар хотира ва вакт нуктаи назаридан киммат булиб, компиляция вактида бажарилувчи натижалари кейинчалик йук килиниши керак булган харакатларни синтаксис анализаторга кушишни кийинлаштиради (масалан, белгилар жадвалини куриш ва х.к.). Бундан сунг биз факат разборнинг детерминирован усуллари хакида суз юритамиз.

Синов саволлари

1. Разбор муаммоси нима?
2. Чап томонли ва унг томонли разбор нима?
3. Нима учун асосий эътибор, жуда куп сонли чап ёки унг томонли булмаган разборлар була туриб, чап ва унг томонли разборларга каратилган?
4. Синтаксис дарахтни куришда разбор муаммосини кандай хал этиш мумкин?
5. Бир кийматли булмаган грамматикага таъриф беринг.
6. Детерминирован ва нодетерминирован разбор усулларини фаркини айтиб беринг
7. Куйидаги тугилувчи коидаларга эга грамматика берилган: S S+T F (S)
   S T F a
   T T*F F b
   T F
   b) (a+b)*a+a ифода учун синтаксис дарахтни куринг.
   c) (a+b)*a+a ифода учун чап томонли разборни куринг.
   d) (a+b)*(a+b) ифода учун унг томонли разборни куринг.
8. Куйидаги тугилувчи коидаларга эга грамматика бир кийматли эмаслигини исботланг.
   S if c then S else S S x
   S if c then S

Маъруза №6. Мавзу: Компиляция босқичлари. Компилятор тузилиши.

Олдин айтилганидек, трансляция жараёни икки босқичдан: тахлил босқичи ва натижавий дастурни ҳосил қилиш жараёнидан иборат[1]. Биз ушбу босқичларни компиляция жараёни мисолида кўриб чиқамиз ва компиялтор қисмлари билан танишамиз. Компиляция жараёнинг умумий схемаси қуйидаги 1-расмда келтирилган.



Расм 4. Компиляция схемаси

Ушбу схемага кўра анализ босқичи 3 та фазадан: лексик анализ, синтаксис анализи ва семантик анализ босқичларидан иборат. Анализ жараёнида дастурнинг ички (оралиқ) кўриниши, идентификаторлар жадвали шакллантирилади. Дастурда учраган хатоликлар анализ қилинади (яъни хатонинг тури аниқланади) ва локализация қилинади (яъни хато учраган модуль ва/ёки сатр аниқланади), сўнгра топилган хатоликлар хақида фойдаланувчига хабар берилади. Натижавий (объектли) кодни синтез (ҳосил) қилиш босқичи икки фазадан: код генерациясига тайёргарлик ва бевосита код генерацияси фазаларидан ташкил топган. Код генерацияси натижасида объектли код ҳосил қилинади.
Юқоридаги схемадан кўриш мумкинки, формал тиллар нуқтаи-назаридан компилятор трансляция жараёнида икки ҳил функцияни бажаради. Бошланғич тил учун компиляторнинг анализ қилувчи қисми фарқловчи вазифасини бажаради, яъни аниқроқ айтадиганда, агар дастур бошланғич тил қоидалари асосида тўғри ёзилган бўлса, у ҳолда

[1] Таҳлил босқичи баъзан анализ босқичи деб ҳам аталади. Натижавий кодни ҳосил қилиш жараёни баъзан синтез, баъзан генерация босқичи ҳам дейилади.

таҳлил муваффақиятли якунланади ва дастурнинг ички (оралиқ) кўриниши ҳосил қилинади. Аксинча бўлса у ҳолда дастурдаги хатоликлар аниқлаштирилади ва бу ҳақда фойдаланувчига хабар берилади. Иккинчи томондан компиляторнинг натижавий тилдаги кодни синтез қилувчи қисми машина тилидаги кодни генерация (ҳосил) қилиш билан шуғулланади, ва бунда албатта у машина тилидаги тўғри дастурларни аниқловчи формал грамматика қоидаларига асосланади. Ушбу ҳосил қилинган дастурлар энди бевосита ижрочи (процессор) томонидан танилади.

Компиляция фазалари билан батафсил танишиб чиқайлик. Аввало шуни айтиш керакки, юқоридаги расмда тасвирланган схема – умумий манзара. Конкрет компиляторда ушбу схема анча ўзгартирилган шаклда учраши мумкин.

Лексик таҳлил. Лексик таҳлил давомида дастур матни чапдан ўнгга томон символма-символ кўриб чиқилади ва символлар лексемаларга бирлаштирилади. Эслатиб ўтамиз дастурнинг маънога эга энг кичик тузилмаси **лексема** деб аталади. Дастурда лексема сифатида калит сўзлар (маъноси олдиндан тайин қилинган идентификатор), идентификаторлар, сонлар, арифметик, мантиқий ёки бошқа турдаги амал белгилари келиши мумкин. Лексик таҳлил давомида компилятор лексемалар рўйхатини қуради. Ушбу рўйхатда лексемма, унинг тури ва бошқа информация сақланади. Шунингдек агар лексема идентификатор бўлса, у ҳолда бу лексема идентификаторлар рўйхатига ҳам қўшилади. Лексик таҳлилчининг ишлаш тартиби қуйидаги 2-расмда тасвирланган.

Расм 5. Лексик таҳлилчининг ишлаш тартиби.

Синтаксис таҳлили. Синтаксис таҳлили учун бошланғич маълумот лексеммалар рўйхати бўлиб ҳисобланади. Синтаксис таҳлили давомида компилятор қиладиган иш – лексеммалар рўйхати асосида тилнинг тўғри синтаксис конструкцияси, масалан бирор оператор, ифода, таърифлаш конструкцияси ва ҳакозолар тўғри қурилганлигини текширишдан иборат. Масалан қуйидаги лексеммалар кетма-кетлигини кўриб чиқайлик:

```
for i:=1 do 5 to end begin;
```

Лексик таҳлилчи ушбу занжирда ҳеч қандай хатоликни пайқамайди, чунки барча лексемалар Pascal тили қоидалари мос, ёки Pascal тилидаги калит сўзлар ҳисобланади. Операторнинг нотўғри ёзилганлиги синтаксис таҳлили пайтида аниқ бўлади. Эслатиб ўтамиз юқоридаги занжирнинг тўғри кўриниши

```
for i:=1 to 5 do begin end;
```

шаклида бўлиши керак. Албатта синтаксис таҳлил пайтида ҳам учраган хатоликлар юқорида айтилган тарзда қайта ишланади.

Семантик таҳлил. Семантик таҳлил давомида бошланғич тилнинг семантик қоидалари кўра дастурнинг тўғрилиги текширилади. Одатда бундай қоидалар синтаксис қоидалари аниқловчи формал грамматика орқали ифодаланмайди[2]. Масалан семантик қоидалар қаторига идентификаторларни ишлатишдан олдин эълон қилинганлиги, арифметик ифодада, қиймат беришда, функция ёки процедура чақирувида аргументларнинг формал параметрларга тип ва сон жиҳатдан мослиги, шартсиз ўтиш операторларнинг тўғри

[2] Семантик қоидаларни ҳам аксарият ҳолларда формал грамматика орқали аниқлаш мумкин. Лекин унда тилнинг синтаксиси ва демак фарқловчининг ҳам мураккаблиги бир-неча баробар ортиб кетади.

қўлланилганлиги[3] текширилади. Семантик таҳлил қисман синтаксис таҳлил фазасида, қисман код генерациясига тайёргарлик фазасида амалга оширилади. Биз семантик таҳлилни код генерацияси билан боғлиқ ҳолда ўрганамиз.

Дастурнинг ички (оралиқ) кўриниши. Дастурнинг ички кўриниши – бу таҳлилнинг турли босқичлари натижасида ҳосил қилинган бир қанча маълумотлар тузилмаси бўлиб, унинг асосини у ёки бу кўринишда қурилган синтаксис таҳлили дарахти ташкил этади. Ушбу дарахт ва идентификаторлар жадвали асосида сўнгра семантик таҳлил, код генерациясига тайёргарлик ва бевосита код генерацияси бажаралади.

Код генерациясига тайёргарлик. Ушбу босқич синтез босқичининг биринчи фазаси бўлиб ҳисобланади. Одатда ушбу босқичда иккита асосий иш бажарилади. 1) код ва маълумотлар учун хотира тақсимоти ва 2) дастурнинг глобал оптимизацияси.

Код генерацияси. Ушбу босқичда дастурнинг оралиқ кўриниши ва олдинги фазада амалга оширилган хотира тақсимоти асосида объектли код генерацияси бажарилади.

Кейинги лекцияларда ушбу босқичлар ҳақида батафсил тўхталамиз.

[3] масалан цикл ташқарисидан goto ёрдамида цикл ичидаги белги қўйилган операторга ўтиш мумкин эмас.

Маъруза №12. Мавзу: Формал тил ва грамматикалар.

Режа:

1. Формал тил ва грамматикалар
2. Белгилар занжири ва улар устидаги амаллар.

Формал тил – бу сузлардан ташкил топган гапларнинг тупламидир (каторлар).

Катор – бу чекланган узунликдаги белгилар кетма-кетлигидир (белгилар бири биридан кейин ёзилган). Ушбу белгиларнинг хар бири аввалдан берилган алфавитнинг булаги хисобланади.

Алфавит – белгиларнинг ёки литераларнинг буш булмаган тугалланган туплами булиб, улар ёрдамида каторларни куриш мумкин. Формал тилда каторлардан гаплар курилади ва бу каторлар белгилар занжири деб аталади.

Изох:

Формал тилни берилиши учун унинг алфавит ива формал граматикасини курсатиш зарур.

Формал грамматика – каторлар тупламини ифодалаш учун керак буладиган коидалар тизимидир (белгиларнинг тугалланган кетма-кетлиги).

Уз навбатида каторлар гапларни ташкил этади ва х.к. Бундай каторлар туплами **тилни** ташкил этади. Тил кандайдир формал грамматика билан ифодаланади деб хисобласак, грамматика тилни юзага келтиради.

Фараз килайлик алфавит бу ={0, 1} булсин. Ушбу алфавитдан 01001 катор ташкил этилсин. Занжирдаги белгилар сони занжирнинг узунлиги m деб аталади (белгиларнинг мос тушишига богликсиз равишда). Буш занжир деб узунлиги m=0 булган занжир аталади. У .. ёки .. белгиланади.

Белги алфавитда аникланган каторлар тупламини англатади.

..={0,1,01,10,101,110,..}.

Буш туплам ихтиёрий тупламга киради.

L алфавитнинг формал тили .. дебкандайдир ихтиёрий.. . кисмтупламга айтилади. Знак .. разница с .. в одном символе. .. не включает в сея пустую строку.

Хар бир формал тил гаплари кандайдир коидалардан келиб чиккан холда тузилади. Бу коидалар **тилнинг синтаксисини** ташкил этади. Формал тилнинг муаммоларидан бири тил синтаксисини ифодалашдан иборатдир. Мазмуни: агар тиллар куп булмаган тугалланган гаплардан ташкил топган булганларида эди, у холда уларни санаб чикиш мумкин булар эди ва конструкцияларнинг тугрилиги ущбу тупламга тегишли ёки тегишли эмаслиги билан текширилар эди.

Лекин мумкин булган гаплар сони шунчалик купки, уларни санаб чикишнинг сира иложи йук.

Шунинг учун тиллар синтаксисини ифодаловчи махсус –**метатиллар** (тиллар устида тиллар) мавжуд. Ушбу тилда тил конструкциясини тугрилигини аникловчи коидалар тизимини ифодаланади.

Грамматика – тил синтаксисини коидалари тупламидир.

Грамматика икки хил куринишда булиши мумкин:

- Тугилувчи ;

- Англовчи .

Тугилувчи грамматика тугри гаплар ташкил этишни процедурасини бошлангич белигидан бошлаб ифодалайди.

Англовчи грамматика эса аник конструкцияни аник тилга тегишли эканлигини англаш жараёнини ифодалайди.(занжирдан бошлангич белгигача).

Изох:

Бу грамматикалар харакат йуналиши жихатидан фаркланадилар.

Тугилувчи грамматикага мисол куриб чикамиз.

Бу процедура рус тилини кесилган булаги куринишини эслатади. Грамматика объектлари: гап аъзолари, гап булаклари.

Белгиланиши:

ПР- предложение -гап

П – подлежащее- эга

С – сказуемое - кесим

ИС – имя существительное – от исми

М – местоимение - олмош

ГФ – глагольная форма – глагол куриниши

Коидалар:

1. <ПР>--<П><С>
2. <П>--<ИС>
3. <П>--<М>
4. <С>--<ГФ>
5. <ИС>--самолет
6. <ИС>--дом
7. <М>--он
8. <ГФ>--стоит
9. <ГФ>--строится
10. <ГФ>--летит

Белгиланиши:

a—b a b ни келтириб чикаради; a дан b чикади; b a дан келиб чикади.

Курсаткич чикиш йуналишини курсатади.

-- бор, кандай аникланади.

Изох:

Унгдаги нарса чапдаги нарсани аниклайди.

<имя> - нетерминал белги

имя – терминал белги.

Терминал белгилар грамматика коидалари ёрдамида очиб берилмайди, нетерминал белгилар эса очилади.

Грамматиканинг коидалар туплами **P** деб аталади.

Ушбу холда **P** санаб утиш куринишида келтирилган**.**

Ёки чап ёки унг кисмга кирувчи, ёки иккала кисмга хам кирувчи барча белгилар туплами **V** билан белгиланади.

P коидада **V** алфавитнинг белгиларидан тилнинг тугри гапларини тузиш хакидаги маълумотлар сакланади. (ушбу коидалар буйича тузилмаган гаплар нотугри хисобланадилар). Ушбу холатда $\mathbf{G_0}$ грамматика учун алфавит куйидагича белгиланади:

V = {<ПР>, <С>, <ИС>, <П>, <М>, <ГФ>, самолет, дом, он, стоит, строится, летит}.

Умумий холатда **V** туплам икки кисмдан **N** ва **T**, белгилар тупламидан тузилади. Ихтиёрий нетерминал белги жуда булмаганда бир маротаба коиданинг чап томонига кириши шарт (**N** туплам).

N = {<ПР>, <П>, <С>, <М>, <ИС>, <ГФ>}

T – факат унг булакка кирадиган терминал белгилар туплами.

T = {самолет, дом, он, строится, стоит, летит}

Тугилувчи грамматики учун **S** – бошлангич белги.

S = {<ПР>}.

Компилятор учун (Паскаль) S сифатида «дастур» тушунчаси туради.

<Программа>--<Раздел описаний><Раздел действий>.

Агар унг булакда бир конструкция кейингисидан кейин катъий келса бу хол каторлар конкатенациясини англатади (занжирланиш) .

Ихтиёрий грамматика иккита масалани хал килиши керак:

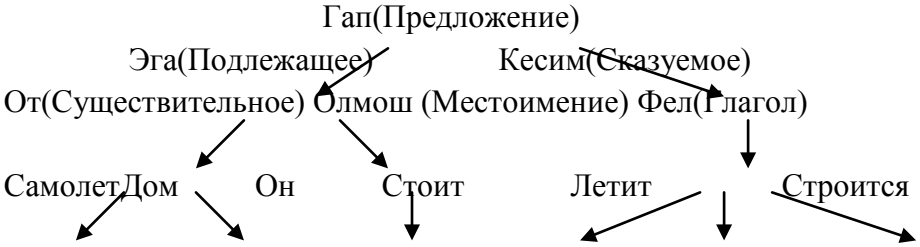
- Англаш масаласини (либо задачи распознавания);
- Ёки тугилиш масаласини (либо задачи порождения).

Тугилиш жараёнида тугри гапларнинг чикиши ифодаланади. Бу шундай амалга оширилади: бошлангич белгидан бошлаб чап булакнинг коидаларини унг булакка алмаштириш амалга оширилади.

Хар бир олинган тушунча узининг тарифига алмаштирилади. Бу жараён унг томонда факат терминал белгиларнинг узи колмагунича давом этади.

Жараённи грамматик разбор дарахти куринишида ифодалаш кулай. Ушбу дарахт кандай коидаларни кандай тил конструкцияларга куллаш мумкинлигини курсатади, лекин у аник тугилиш жараёнидаги куллаш тиртибини курсатмайди. Дарахт грамматиканинг ушбу коидаларига асосланган холда курилади. Юкорида бошлангич белги жойлашади, пастда – терминал белгилар. Дарахт №1 коидаларни куллаш йули билан курилади.

Гап эга ва кесимнинг конкатенациясидан ташкил топади. Кесим булиб ёки ИС, ёки М келиши мумкин.



Аник бир гапни чикиши ечим кабул килишни талаб этади: кайси йул билан пастга караб юриш керак. Дарахт эса коидаларни ифодалайди. Самолет строится – масалан.

Англаш масаласи дарахтдан фойдаланиб ечилади.

Дарахт буйича пастдан юкорига харакат килиб аник гапга бошлангич белгига етиб бориш керак. Бу ерда унг томон булакларини чап томон булаклари коидасига алмаштирилади.

Масалан, Дом летит.

ИС ва ГФ; --П ва С; --ПР.

Умумий холда грамматика :

1. Нетерминал белгилар туплами.
2. Теминал белгилар туплами.

3. Бошлангич белги.
4. Коидалар тупламидан ташкил топади.

$G = \{N, T, S, P\}$

Коидалар куйидаги куринишга эга: $a \rightarrow b$

$a \in (N \cup T)^+$

$b \supset (N \cup T)^*$

+ бу тупламга буш туплам киритиш мумкин эмаслигини англатади

* бу тупламга буш туплам киритиш мумкинлигини англатади

a ва b – баъзи бир каторлар (белгилар кетма-кетлиги)

Бундай коидалар **продукциялар** деб хам аталади.

G1 грамматикага мисол караб чикамиз. Бу холда куйидагилар киритилади:

Нетерминал белгилар $\rightarrow$ A, B, C, …

терминал белгилар $\rightarrow$ a, b, c, …

Караймиз

$G1 = \{N, T, S, P\}$, бу ерда

$N = \{A, B, S\}$

$T = \{a, b\}$

$P = \{ S \rightarrow AB$ (1)

$A \rightarrow aA$ (2)

$A \rightarrow a$ (3)

$B \rightarrow Bb$ (4)

$B \rightarrow b$ (5)

}

№2 коидада нетерминал белги А хам чап хам унг булакда мавжуд. Бу эса А белги тегишли каторлар синфи А белгига а префиксни кушиш оркали курилишини англатади.

Учинчи коидада А а оркали аникланади. Умумий холда бундай жараён каторлар конкатенацияси деб аталади (А катор чапдан кушиладиган а конкатенациясидан курилади).

Кандайдир тушунча узи узидан куриладиган холат рекурсия деб аталади.

Туртинчи коидада хам рекурсия мавжуд. Занжир белгини унгдан кушиш йули билан ташкил этилади.

Бу холатда коидалар барча мумкин булган вариантларни санаб чикиш йули билан берилади, хар бир вариант учун бита катор.

Грамматика коидаларини бериш усули нотация деб аталади.

Купинча Бэкус Наура формасидан фойдаланилади.Унда куйидаги белгилашлардан фойдаланилади: $\rightarrow$; ::=

Барча нетерминал белгилар бурчак кавсларга олинади. Агар кандайдир тушунча учун чап булакда бир нечта вариант бор булса, у холда “|” белгидан фойдаланилади.

<A> ::= a | a<A>

Бундай ташкари грамматика коидалари метабелгилардан фойдаланилган холда хам берилади (кавслар):

( ) – думалок кавсларда санаб утилган барча конструкциялардан ушбу конструкция учраган вактида факат биттагинасидан фойдаланилади.Вергулдан булаклаш учун фойдаланилади.

[ ] – бу кавсларга киритилганлар булиш хам, булмасликлари хам мумкин.

{ } – такрорлашни англатади (n марта, n = 0,1,2,…. 0 конструкция мавжуд эмаслигини англатади.)

Метабелгилик ва метабелгисиз ишоралик бутун сонлар учун коидаларни берилишини караб чикамиз.

G = {{0,1,2,…,9},{<ишорали сон>,<сон>,<ракам>}, P,<ишорали сон>}

P = {

<ишорали сон> ::= +<сон> | -<сон>

<сон> ::= <ракам> | <ракам><сон>

<ракам> ::= 0|1|2|3|4|5|6|7|8|9

}

Иккинчи коида рекурсив.

Худди шунингдек метабелгилик ишоралик сонлардан фойдаланилган хол учун :

< ишорали сон > ::= [(+,-)] ракам{ракам}

2.Белгилар занжири ва улар устидаги амаллар.

Белгилар занжири – бу бири биридан кейин ёзилган белгиларнинг ихтиёрий кетма-кетлигидир.

Белгилар занжири (БЗ) учун таркиб, белгилар сони ва тартиб мухимдир. БЗ α ва β тенг α=β ёки мос тушадилар, агар улар битта белгилар таркибига эга булсалар, бир хил белгилар сонига ва белгиларнинг занжир буйлаб бир хил келиш тартибига эга булсалар. Занжирдаги белгилар сони занжир узунлиги дейилади.

БЗ куйидаги хусусиятларга эгадирлар:

1)Конкатенация – 2 та занжирни йигиндиси ёки купайтмаси αβ

α=“ВА”

β=“CL” => αβ=“BACL”

Конкатенация амали коммутиция хусусиятига эга эмас, яъни αβ≠βα. Ассоциативлик хусусиятига эгадир (αβ)γ=α(βγ)

2)Занжирга мурожат – занжир белгиларини тескари тартибда ёзиш α^R, α=“ВАСЯ” => α^R=“ЯСАВ” Ушбу амал учун $(\alpha\beta)^R=\alpha^R\beta^R$ хакикат

3)якинлашув – занжирни n марта такрорлаш

4)белгиларнинг буш занжири – бу битта хам белгига эга булмаган занжирдир, λ -буш занжир учун куйидаги хакикат: 1) |λ|=0; 2) ихтиёрий α: λα=αλ=α; 3)λ^R=λ; 4) ихтиёрий n≥0:λ^n=λ; 5)ихтиёрий α: α^0=λ

Назорат саволлари

1. Тил синтаксисини нималар аниклайди?
2. Тилни синтаксиси ва семантикаси орасида кандай фарк бор?
3. Грамматика нима ва у кандай берилади?
4. Тилнинг терминал ва нотерминал белгилари кандай фаркланадилар?
5. У ёки бу белгининг тилнинг гапларида учрашини кандай изохлайсиз?
6. «Бошлангич белги» нима ва у тилнинг бошка белгиларидан нима билан фарк килади?
7. Грамматика тугилувчи коидалар оркали берилган булсин.

S ->(S) S->E,

S->SS бу ерда S – бошлангич белгиб, E буш катордир.

Куйидаги каторлар ушбу грамматика буйича генерация килинган тилга тегишлими? Жавобингизни исботланг.

а) катор((1)()), в) катор (()()())

Фойдаланилган адабиётлар

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. –СПб: Питер, 2003.-396 с.
2. Афанасьев А.Н. Формальные языки и грамматики: Учебная школа: УлГТУ, 1997. – 84 с
3. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции -: Мир, 1979.-487с.
4. Компаниец Р.И. Системное программирование. Основы построения трансляторов. СПб.:Корна принт., 2000. -256 стр.
5. Дьяконов В.Ю. Системное программирование. Высш.шк.. 1990. -221 с.

Маъруза №13. Мавзу: Формал тил ва грамматикаларнинг классификацияси.

Режа:

1. Бэкус-Наур формасидаги грамматика

Бэкус-Наур формасидаги грамматиканинг ёзилиши.

Куйидаги ёзув куриниши:

1)$\alpha\rightarrow\beta_1$, $\alpha\rightarrow\beta_2$, …, $\alpha\rightarrow\beta_n$

$\alpha\rightarrow\beta_1|\ \beta_2|\ldots\ |\beta_n$

2)барча нотерминал белгилар <A> бурчак кавсларга олинади.

метабелгилардан фойдаланиб грамматика коидаларини ёзиш.

1.() – барча санаб утилганлардан факат биттасигина туриши мумкин.

2.[] – ушбу курсатилган занжирлардан учраши хам мумкин учрамаслиги хам мумкин.

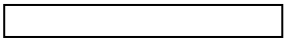
3.{} – ушбу кавсларда келтирилганлар учрамасликлари хам мумкин, ёки 1 марта учрашлари, ёки бир неча марта учрашлари мумкин.

4. , - () кавс ичидаги белгилар занжирини ажратиш учун фойдаланилади.

5. “ ” качонки метабелгилардан бирини занжирга оддий усул билан кушиш керак булганда фойдаланилади.

Грамматика коидаларини граф куринишида ёзиш.

Хар бир нотерминал белгига йуналтирилган граф куринишидаги диаграмма мос келади.

Номланиши	Белгиланиши	Йуналтирилиши
Кириш нуктаси	Хеч кандай белгиланмайди	Ундан графнинг кирувчи кобиги бошланади.
Нотерм белги		Ичида нотерминал белгиларнинг белгиланиши келтирилган
Терминал белгилар занжири		Унда терминал белгилар занжири ёзилган
Боглаш нуктаси		Чорраха
Кириш нуктаси	Хеч кандай белгиланмайди	Унга графни чикувчи кобиги киради

Маъруза №14. Мавзу: Тил синтаксиси ва семантикаси.

Режа:

1. Тил таърифи. Синтаксис ва семантика

1.Тил таърифи. Синтаксис ва семантика

Компиляторни ташкил этишдан аввал киритилаётган тилнинг аник таърифига эга булиш керак.

Бир неча каторлардан таркиб топган тилни тассаввур килишимиз мумкин. Тилни ифодалашда кандай каторлар ушбу тилга тегишли эканлиги (тил синтаксиси) ва ушбу каторларни киймати (тил семантикаси) аникланади. **Синтаксис** - формал тугри гаплар тупламининг коидалари тупламидир. Тилга тегишли каторларни тилнинг гаплари деб аталади. Реал тилларда чексиз гаплар сони булади ва уларни санаб утишнинг иложи йук. Энг содда тилнинг синтаксисини табиий тилда куйидагича ифодалаш мумкин, масалан: «барча каторлар, факат 1 ва 0 лардан ташкил топган» у холда 1111 ва 1000110 –тилга тегишли, 1020 эса йук.

Масалан, куйидаги гап «машина юради». «Машина» сузи эга, «юради» кесим. Ушбу гап куйидаги синтаксис коидалар ёрдамида ифодалаш мумкин булган тилга тегишли:

<гап>::=<эга><кесим>

<эга>::=машина | от

<кесим>::= юради | чопади

Ушбу учта каторнинг маъноси куйидагича: гап эга ва кесимдан иборат. Эга ёки машина деган бир суздан ёки от деган суздан ташкил топган. Кесим хам ёки юради деган суздан, ёки чопади деган суздан ташкил топган.

Ихтиёрий гапни бошлангич белги оркали кетма-кет куйиш йули билан олиш мумкин.

Ушбу коидаларни ёзишда фойдаланиладиган нотация **Бэкус-Наур формаси** деб аталади. Синтаксис бирликлар <гап> <эга> ва <кесим> нотерминал белгилар деб аталади, “машина”, ”от”, ”юради”, чопади терминал белгилар деб аталади, коидалар эса **тугилувчи коидалардир.** ::=, | . <> белгилар метабелгилардир. **Семантика** тилнинг барча гапларига киймат беради.

Алфавит –белгилар туплами. Масалан: Рус харфлари. Лотин харфлари , ракамлар.

Агар А-алфавит булса, А* А га кирувчи барча белгилардан тузилган каторларнинг (буш каторни хам кушган холда) тупламини англатади. А+ эса А га кирувчи барча белгилардан тузилган каторларнинг (буш каторни хам кушмаган холда) тупламини англатади. Буш катор купинча Е(эпсилон) ёрдамида белгиланади

Тилни синтаксисини тупламларни тасвирлаш оркали аниклаш мумкин, масалан L={0n1n|n>=0}. Ушбу тил бир ёки бир неча нуллардан, бирлардан ва буш катордан ташкил топган каторларни уз ичига олади.

Тилни мураккаброк синтаксисини грамматика ёрдамида аниклаш яхширок. Грамматикага тилни гапларини тузиш учун коидалар туплами киради. L синтаксисни оламиз ва куйидаги коидалардан фойдаланамиз.

1. S --> 0S1 2.S --> E

Ушбу тилнинг гапларини чикариш учун куйидагича иш юритамиз. S белгидан бошлаймиз ва уни 0S1 билан алмаштирамиз ёки E билан. Агар S яна олинган каторда мавжуд булса, яна алмаштирамиз ва х.к. Шундай усул билан олинган S га эга булмаган катор шу тилнинг гапи хисобланади. Масалан, S 0S1 00S11 000S111 000111

Бундай каторларнинг кетма-кетлиги 000111 каторни чикиши дейилади, стрелка белгиси эса чикиш кадамларини булаклаш учун хизмат килади. Ушбу тилнинг барча гапларини иккита коидадан келиб чиккан холда келтириб чикариш мумкин, Ихтиёрий келтириб чикариш мумкин булмаган катор ушбу тилнинг гапи хисобланмайди. Грамматикани купинча кайта ёзиш тизими деб хам атайдилар.

Грамматика (Vt,Vn,P,S) туртлик билан аникланади, Бу ерда Vt –алфавит булиб, унинг белгилари терминаллар деб аталади, улардан грамматика оркали келтирилувчи занжирлар курилади. Vn –алфавит булиб, унинг белгилари нотерминаллар деб аталади, занжирларни куришда фойдаланилади. Vt ва Vn умумий белгиларга эга эмаслар, яъни Vt Vn =0, Грамматиканинг тулик алфавити V Vt U Vn каби аникланади.

P – тугилувчи коидалар туплами булиб, унинг хар бир элементи (a, b) жуфтлигидан ташкил топади, бу ерда a V+ да, ва b эса V* да.

a коиданинг чап булаги, b эса унг булагидир. Коида куйидагича ёзилади: a b. S Vn га тегишли ва **бошлангич белги (аксиома)**деб аталади. Бу белги тилнинг ихтиёрий гапини олиш учун таянч нуктадир.

L={0n1n|n>=0}.тилни генерация киладиган грамматика булиб

G0 =({0,1},{S},P,S), ,бу ерда P={S 0S1.S E} хисобланади.

L={anbm|n,m>=0}.тилни генерация киладиган грамматика булиб

G0 =({a,b},{S,A,B},P,S), бу ерда P={S AB, A aA, A E, B bB, B E} хисобланади.

S белгидан бошлаб нотерминални алмаштириш коидасини куллаб aaabb каторни генерация килиш мумкин.

S --> AB --> aAB --> aaAB --> aaaAB --> aaaB --> aaabB--> aaabbB --> aaabb

Хар бир бошлангич белгидан келтириб чикариладиган катор сентенциал форма деб аталади. Сентенциал формали гап –бу факат терминаллардан иборатдир. Терминалларни кичкина харфлар билан, нотерминалларни катта харфлар билан белгилаймиз.

Иккита бир хил тилни келтириб чикарадиган грамматикани **эквивалент грамматика** деб атаймиз.

Маъруза №16. Мавзу: Кодни генерациялаш усуллари. Кодни оптималлаштириш. Дастурларнинг ички кўриниши ва уларни шакллантириш усуллари.

Режа:

1. Кодни оптималлашнинг умумий тамойиллари.

2. Кодни генерациялаш усуллари.

3. Кодни генерациялашда «тўртлик»ни қўллаш

Калит сузлар.

- **Семантик тахлил**
- **Компилятор**
- **Кирувчи занжир**
- **Операнд**
- **Функциялар**
- **Процедура**
- **Параметр**

1.Кодни оптималлашнинг умумий тамойиллари.

Оптимизация бу фойдалирок натижали объект дастур олиш максадида компьютер дастуридаги амалларнинг узгартириш ва тартибга солиш билан боглик кайта ишлашдир. Оптимизация бир неча марта бажарилиши мумкин, код генерациясини тайёрлаш фазаси буйича ва кодни генерациялаш фазаси буйича.Натижавий дастурннинг фойдалилик курсаткичи булиб куйидаги критерийлардан фойдаланилади: 1)натижавий дастурнинг бажарилиши учун зарур булган хотира хажми 2) дастурнинг бажарилиш тезлиги.

Айлантиришларни оптималлашни икки асосий куринишини фарклайдилар: 1) кирувчи дастур матнини натижавий объект кодига боглик булмаган холда унинг ички тасвирланишини куринишида айлантириш. 2) берилган айлантиришлар максадли хисоблаш тизимининг архитектурасидан боглик эмас. Улар аввалдан яхши таниш булган математик ва мантикий айлантиришларга асосланган. 3) натижавий объект дастурни айлантириш.

Ушбу гурух айлантиришлари максадли хисоблаш тизимининг архитектурасидан боглик.

Оптималлаш куйидаги синтаксис конструкциялар учун бажарилиши мумкин: 1)дастурнинг чизикли булаклари; 2) мантикий ифодалар 3)цикллар 4)процедура фукцияларини чакириклари

2.Кодни генерациялаш усуллари.

Кодни ички ёзувларининг бир хил фрагментлари (постфикс ёзувлари амаллари, туртлик ва бошкалар) машина тилининг бир хил буйрукларини ифодалайди. Масалан, код генерацияланаётган PLUS_OP туртлик, агар процессорда барча амаллар регистр-аккумлятор устида бажарилса, хар доим куйидаги кодни генерациялайди:

LOAD регистр , операнд 1

ADD регистр, операнд 2

STORE регистр, натижа

Машина командаларининг бу кетма-кетлиги коррект, лекин оптимал эмас. Масалан, куйидаги гап

X:=X+Y*Z олтита команда оркали амалга оширилади:

LOAD регистр, Y (туртлик (MULT_OP,Sy,Sz,T1))
MUL регистр, Z
STORE регистр, T1
LOAD регистр, X (туртлик (ADD_OP,Sx,T1,Sx))
ADD регистр, T1
STORE регистр, X
Худди шунингдек, ушбу натижага келтирувчи куидаги дастурни куриш мумкин.
LOAD регистр, Y
MUL регистр, Z
ADD регистр, X
STORE регистр, X

Ушбу усул билан генерацияланаётган код хар доим тугри хисобланади, лекин хар доим хам оптимал эмас. Шунинг учун кодни хисоблашларни аниклигига таъсир курсатмай туриб, узгартириш имконини берувчи курилмаларга эга булиш керак.

Хар бир туртликка купгина холларда ягона машина командалари кетма-кетлиги мос келади, код генератори купинча хар бир туртлик буйича кисмдастурлар туплами мос келади.

3. Тўртлик формаси. Кодни генерациялашда «тўртлик»ни қўллаш

Постфикс ёзувдан дастур кодини куриш мумкин, лекин бундай ёзув формасини оптималлаштириш мураккаб иш. Купгина компильяторлар дасурнинг объект кодини куриш учун оптималлаш учун кулай булган ички формалардан фойдаланадилар. Генерация килинаётган коднинг энг куп таркалган ички тасвирлашни формаларидан бири бу **туртликдир.**

Туртлик –бу туртта элементдан ташкил топган объектдир: амаллар, иккита операнд ва натижалар. Агар амал бажариш натижасида кандайдир узгарувчини киймати хисобланса, у холда бундай туртликни куриш унчалик мураккаб эмас. Масалан: X:= Y+Z гап куйидаги туртлик оркали ифодаланади.
(PLUS_OP, Sy, Sz, Sx), бу ифода Sy белгилар жадвали билан ячейкада аникланган узгарувчини Sz ячейкада аникланган узгарувчи билан (PLUS_OP) кушиб ва натижани Sx ячейкада саклашни англатади. Энди бошлангич гап мустакил бирлик сифатида ифодаланади, уни код генератори жойлашиш манзилидан катьи назар кайта ишлай олади. Шундай килиб, оптимизатор амаллар кетма-кетлигини кодни генерациялаш жараёнини мураккаблаштирмасдан узгартириши мумкин.

Унар операторлар учун туртликнинг иккинчи операндини майдонини игнорироват килиш мумкин, иккитадан ортик операндларни талаб киладиган амалларни эса бир неча туртликлардан ташкил топган кетма-кетликлар куринишида ифодалаш мумкин.

Масалан, куйидаги операторни X:= F(A,B,C,D) учта туртлик куринишидаги гурух сифатида ёзиш мумкин.

(F1,A,B, T1)
(F2, T1,C, T2)
(F, T2, D, X)

F1 ва F2 функциялар оралик хисоблашларни амалга оширадилар, T1 ва T2 ячейкалар эса ушбу харакатларнинг натижаларини саклаш учун мулжалланган. Дастурни фактик куриш

вактида код генератори объект кодида F1 (бу амал учун F2 ва F амаллар оркали) амални тугри ифодалаши мумкин.

Яна оралик ячейкалардан фойдаланишга боглик мисол караймиз.

Фараз килайлик куйидаги постфикс ёзувли гап берилган булсин.

SxSxSySz*+:=

Бу гапга Y ва Z купайтириш амаллари, натижани X билан кушувчи ва X узгарувчига олинган суммани узлаштириувчи амаллар киради. Туртликни генерациялаш вактида купайтиришни амалга оширувчи учун ушбу оралик натижани сакловчи ячейка керак булади. Бу холатда, вактинчалик ва ички узгарувчи ташкил этилади деб фараз киламиз. Шундай килиб, каралаётган гап куйидаги кетма-кетликда ифодаланади.

(MULT_OP, Sy, Sz,T1)

(ADD_OP,Sx,T1,Sx),

Бу ерда биринчи туртлик Y ни Z га купайтириш ва натижани T1 ячейкага ёзишни, иккинчи туртлик эса X узгарувчини Y*Z амал натижасини сакловчи T1 узгарувчи билан кушишни ва суммани X га ёзувни аниклайди.

Назорат саволлари.

1. Семантик тахлил босқичининг вазифаси нмалардан иборат?
2. Семантик тахлил босқичлари хақида маълумот беринг.
3. Кодни оптималлашнинг қандай усулларини биласиз?

Фойдаланилган адабиётлар

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. –СПб: Питер, 2003.-396 с.
2. Афанасьев А.Н. Формальные языки и грамматики: Учебная школа: УлГТУ, 1997. – 84 с
3. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции -: Мир, 1979.-487с.
4. А.Левин. Самоучитель полезных программ. Питер. Санкт-Петербург, 2002.
5. Карпов Б.И. Delphi: Специальный справочник. – СПб: Питер, 2001-648с.
6. Карпов Б.И. Visual Basic Специальный справочник. – СПб: Питер, 2000-415с.
7. Карпов С.Ю. Теория автоматов. Учебные пособия для вузов. –СПб: Питер, 2003.-201с.