

УНИВЕРЗИТЕТ У ПРИШТИНИ - К. МИТРОВИЦА
УЧИТЕЉСКИ ФАКУЛТЕТ

др Синиша Минић

ОСНОВИ ИНФОРМАТИКЕ
И РАЧУНАРСТВА

ЛЕПОСАВИЋ, 2005.

6.2 Режи́ми рада рачунара

Програми се у процесору рачунара могу извршавати на неколико различитих начина, као што је приказано на Слици 6.3, које називамо Режи́ми обраде или Режи́ми рада. Пре свега постоје једно-програмски и вишепрограмски (мултипрограмски) режим рада и паралелна обрада (мултипроцесирање).

Рачунарски систем ради у једнопрограмском или монопрограмском режиму ако процесор опслужује само једног корисника. При томе се у оперативној меморији рачунара налази само један програм који се извршава од почетка до краја. Рачунар у овом режиму опслужује кориснике на два начина:

- (а) појединачном обрадом (индивидуално опслуживање) и
- (б) пакетном обрадом (индиректно опслуживање).

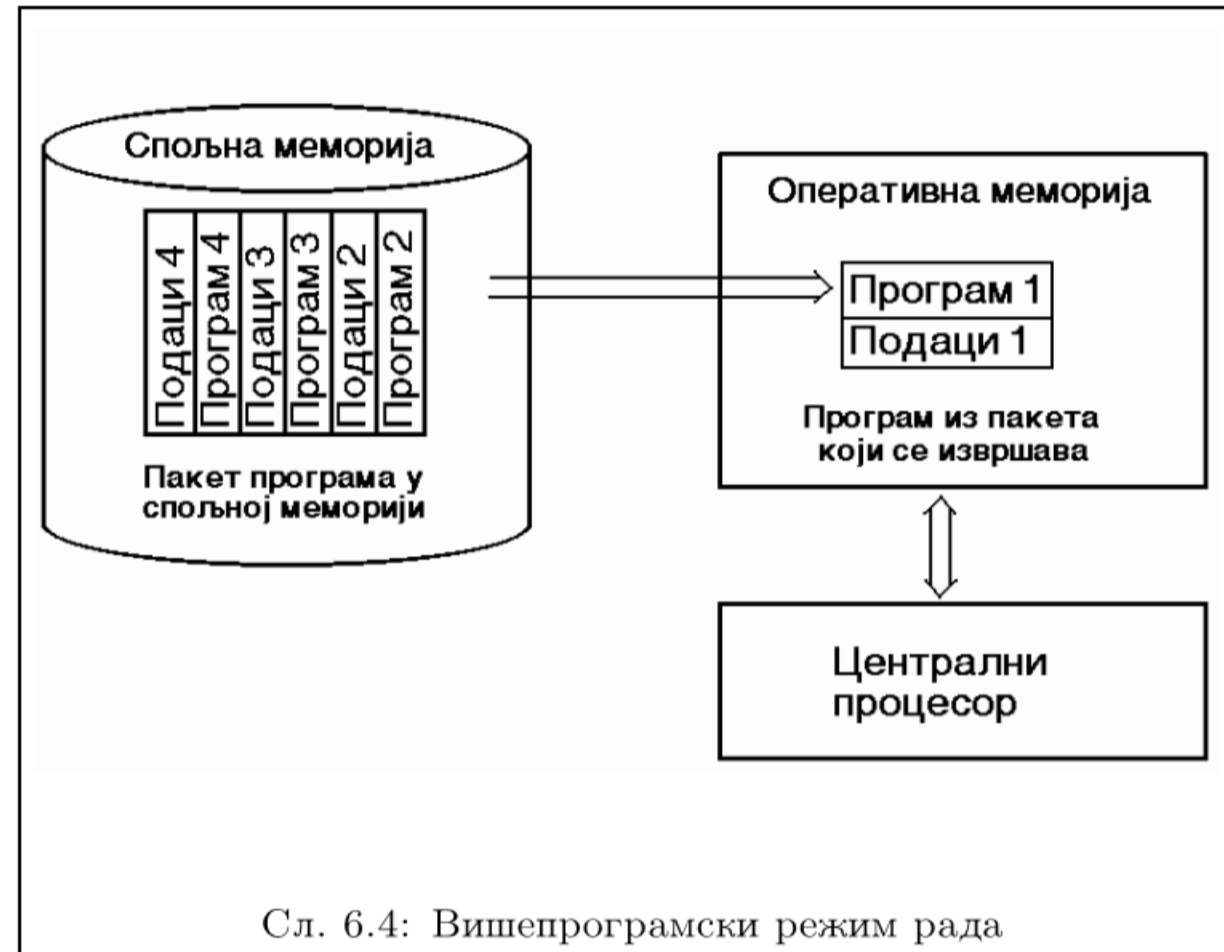
Код појединачне обраде рачунар је у процесу решавања проблема у потпуности на располагању једном кориснику. Корисник сам ради са рачунаром, уноси програм или га позива са спољне меморије, стартује извршење програма, чиме приступа решавању свог проблема и чека резултате. Овакво опслуживање корисника било је у првој генерацији и добрим делом у другој генерацији рачунара, а данас се користи код персоналних рачунара.

Основни Режи́ми рада
<ul style="list-style-type: none"> • Једнопрограмска обрада (монопрограмска обрада) • Вишепрограмска обрада (мултипрограмска обрада) • Паралелна обрада (мултипроцесирање)
Једнопрограмска обрада
<ul style="list-style-type: none"> • Појединачна обрада (индивидуално опслуживање) • Пакетна обрада (индиректно опслуживање)
Вишепрограмска обрада
<ul style="list-style-type: none"> • Обрада са поделом времена • Рад у реалном времену • Даљинска обрада • Дистрибуирана обрада

Сл. 6.3: Режи́ми рада рачунарских система

Пакетна обрада, или, тачније речено, једнопрограмски пакетни

режим рада, јесте начин рада рачунара у коме се обрађују пакети. Пакет (енгл. batch) представља групу програма и одговарајућих улазних података која се за обраду на рачунару третира као једна целина. Пакет је смештен на спољној меморији као ред чекања из кога се програми читају и обрађују један по један редоследом у коме се налазе у пакету (сл. 6.4).



Сл. 6.4: Вишепрограмски режим рада

Код пакетне обраде корисник нема непосредан приступ рачунарском систему. Припремљени програми и подаци предају се оператеру рачунара који их записује на спољну меморију где се формира пакет или корисник са свог терминала смешта програме и податке у пакет на диску (спољној меморији). Извршавањем програма код пакетне обраде управља оперативни систем који обезбеђује прелаз са једног програма на други и контролише њихово извршавање. Обрада програма је секвенцијална (серијска). У једном временском тренутку обрађује се само један програм, док се остали програми налазе у стању чекања. За време извршења једног програма пакета забрањено је његово прекидање. Прелаз на следећи програм могућ је само ако је завршен претходни или је он прекинут због неке изузетне ситуације, као што је грешка у програму и сл.

Резултати обраде памте се на диску да би се проследили кориснику у тренутку када се ослободе излазне јединице, на пример

штампач, или када се ослободи терминал.

Велики број комерцијалних система користио је пакетну обраду за тзв. обраду промена или обраду трансакција. Улазни записи, обично називани записи промена или записи трансакција, прво су прикупљани на спољној меморији онако како су они пристизали, а онда су обрађивани сви заједно. Сви улазни записи формирани у току одређеног временског периода груписани су у посебну датотеку која се назива датотека промена. Датотека промена се затим обрађује ради ажурирања података у основној - матичној датотеци.

Мана обраде промена је у томе што матична датотека и њени записи никада нису сасвим ажурни. Међутим, ова обрада се и даље користи, чак и на данашњим бржим и сложенијим рачунарима, јер у многим случајевима нема потребе да се сваког момента располаже најажурнијим информацијама, на пример при обрачунању потрошње електричне енергије.

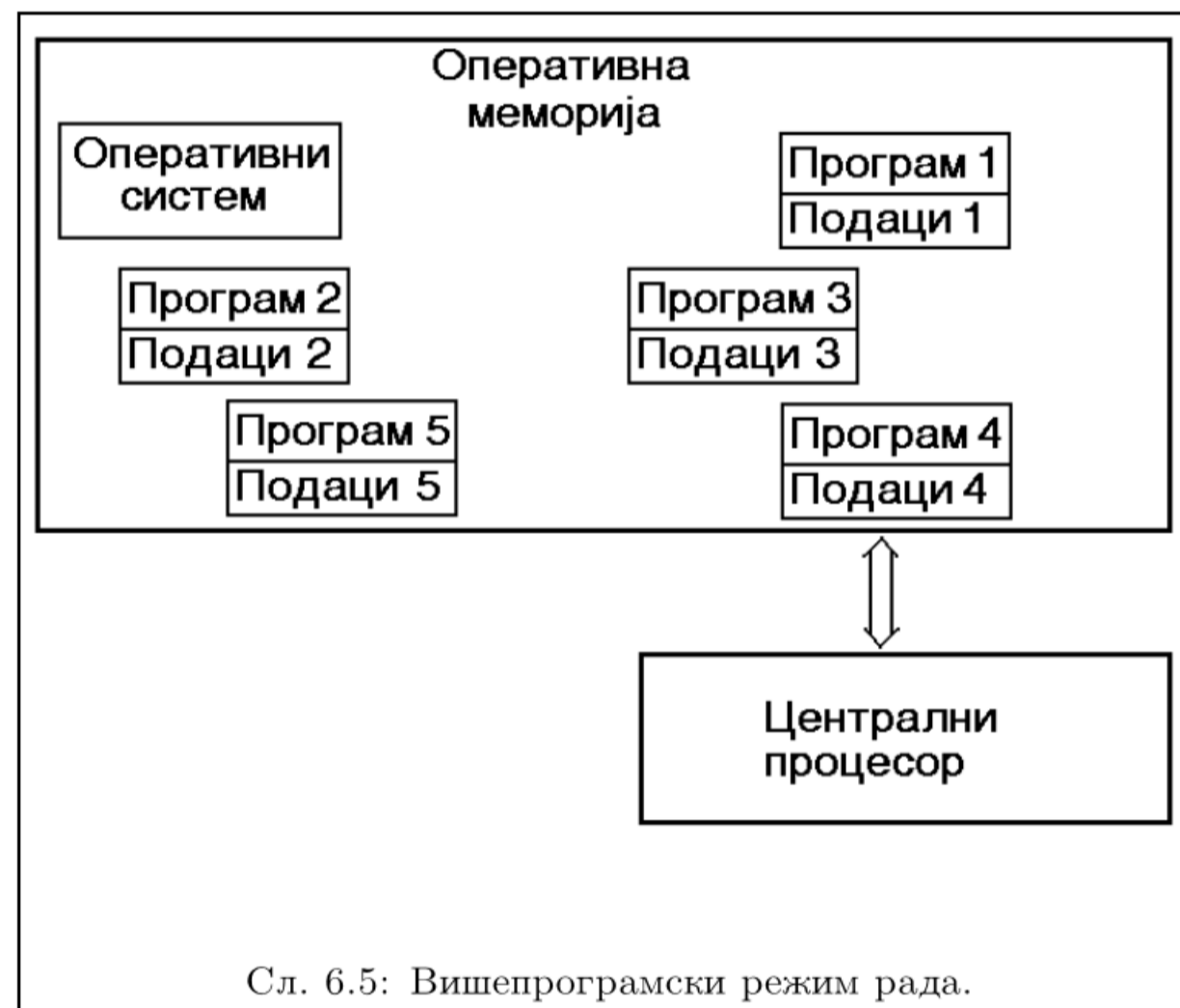
Степен ефикасног коришћења рачунара код једнопрограмске и пакетне обраде је био низак. Обрада се изводила серијски, то јест требало је сваки поступак довршити да би следећи могао да отпочне. Ако процесор мора да се бави прво читањем и памћењем података, затим њиховом обрадом и, на крају, издавањем резултата за, рецимо, штампач, већина његовог времена отпада на улазно-излазне процесе. Тако је удео саме обраде у укупном времену био врло мали. То је последица, с једне стране, веома малих брзина улазно-излазног преноса података и, с друге стране, изузетно велике брзине њихове обраде у процесору.

Систем који садржи већи број процесора и сви они могу учествовати истовремено у решавању једног истог проблема назива се вишепроцесорски рачунарски систем, мултипроцесор или паралелни рачунар. При томе се вишепроцесорским системом не сматра систем чији додатни процесори служе само за помоћне, рецимо улазно-излазне или телекомуникационе активности. Обрада на оваквим системима назива се паралелна обрада или мултипроцесирање.

Чинилац који ограничава степен ефикасности рада централног процесора је сразмерно мала брзина рада улазних и излазних уређаја. Ако у току рада централног процесора на једном програму има доста празног хода због принудног чекања на улазне податке, може се у оперативну меморију сместити и активирати два или

више програма истовремено. Када процесор не може да ради на једном од њих, прећи ће на обраду другог. Овај начин рада назива се вишепрограмски начин рада или мултипрограмирање.

Рачунарски систем ради у вишепрограмском начину рада ако се више корисничких програма налази у оперативној меморији, сви су они активирани за извршење и извршавају се по одређеном редоследу (Слика 6.5). Тиме је омогућена конкурентна ("паралелна", "истовремена") обрада више програма. При томе програм који се извршава може бити прекинут ради преласка на извршење неког другог програма, а накнадно се може вратити на тај исти програм. Мултипрограмирањем се побољшава искоришћавање процесора, као и оперативне меморије.



Као и у случају једнопрограмског начина рада, и код вишепрограмског начина рада процесор ради строго серијски извршавајући инструкцију по инструкцију. Када се говори о конкурентном извршавању програма, има се у виду да после извршавања дела једног програма процесор прелази на извршавање дела другог програма, затим следећег итд., задржавајући при томе потребне информације за могући повратак и настављање обраде једног од претходних програма. Овај начин обраде обезбеђује значајно повећање свеукупне

ефикасности процесора, мада је обрада појединачних програма, по правилу, дужа него код једнопрограмских рачунара.

Мултипрограмирање се може обављати само на рачунарима који имају сложен оперативни систем који је у стању да доноси одлуке о томе како ће се који програм позивати. Други предуслов је довољан број периферија у директној вези с рачунаром које могу истовремено обављати своје послове. Одлука о редоследу извршавања програма доноси се према приоритетима саопштеним оперативном систему на почетку рада.

Теоретски, мултипрограмирањем је могуће постићи ефикасност централног процесора од 100%. У пракси је мало вероватно да ће след наизменичне обраде програма тећи без прекида, па процесор и даље није потпуно искоришћен. Ипак, овај начин обраде и те како повећава његову ефикасност.

Пакетна обрада се може организовати и код вишепрограмског начина рада. То су вишепрограмски системи са пакетном обрадом.

Основни проблем вишепрограмског начина рада је организација заштите од утицаја једног програма на други. Нарочито је важна заштита оперативне меморије, која се састоји у томе да један програм не срне приступати деловима оперативне меморије додељеним другим програмима или оперативном систему.

У оквиру вишепрограмског начина рада разликује се неколико карактеристичних врста обраде, а то су:

- обрада са поделом времена,
- рад у реалном времену,
- даљинска обрада,
- дистрибуирана обрада.

6.2.1 Обрада са поделом времена

Суштина начина рада са поделом времена (или временском поделом) јесте да се обезбеди да више корисника готово истовремено има приступ рачунару. Корисници располажу удаљеним терминалима или персоналним рачунарима који могу обављати и улогу терминала, повезаним телекомуникационим везама са рачунаром. Сваки је терминал снабдевен властитом меморијом, у којој се подаци држе у току улазних и излазних операција. Овде време

одзива, тј. време које је потребно рачунару да започне слање информација терминалу, мора бити кратко. Оно је, у ствари, обично краће од десетак секунди.

Подела времена представља такво распоређивање задатака (процеса) где је задатку дозвољено да се извршава током унапред дефинисаног периода времена, пре поновног распоређивања. У подели времена више корисника истовремено приступа рачунару.

У пракси се поступа тако што се сваком корисничком терминалу додељује унапред дефинисани период времена у току кога се његов програм ставља на располагање централном процесору. Тај период времена је кратак, неколико десетина милисекунди. Због релативно брзог одзива система сваки корисник има утисак да ради сам са рачунаром. Ова истовременост претпоставља да:

1. сваки корисник поседује сопствене уређаје за приступ рачунару;
2. при истовременом приступу више корисника рачунар мора да се понаша, са тачке гледишта корисника, на исти начин као и при индивидуалном опслуживању.

За опслуживање у подели времена користе се вишепрограмски системи. Разликују се три основна режима опслуживања корисника у подели времена:

- (а) режим дијалога (интерактивни режим),
- (б) трансакциони режим и
- (в) универзални режим.

У режиму дијалога корисник са свог терминала има директан приступ рачунарском систему и може поставити један или више упита. Када се на једном од терминала генерише упит, оперативни систем прекида извршење текућег програма и у меморију позива програм намењен обради упита. Код трансакционе обраде врши се расподела рачунарских ресурса између одређеног броја удаљених корисника који преко сопствених терминала директно приступају рачунарском систему. За разлику од пакетне обраде трансакција, овде се ажурирање матичне датотеке обавља одмах. Рачунарски систем са радом у универзалном режиму може радити у било ком од описаних режима. Пример система са временском поделом су

системи за резервацију авионских карата: пошто се преко терминала унесе захтев за место у авиону на одређеном лету, појавиће се списак свих слободних седишта. Резервисано место мора се одмах записати, то јест подаци ажурирати како следећи путник не би добио нетачну информацију.

6.2.2 Рад у реалном времену

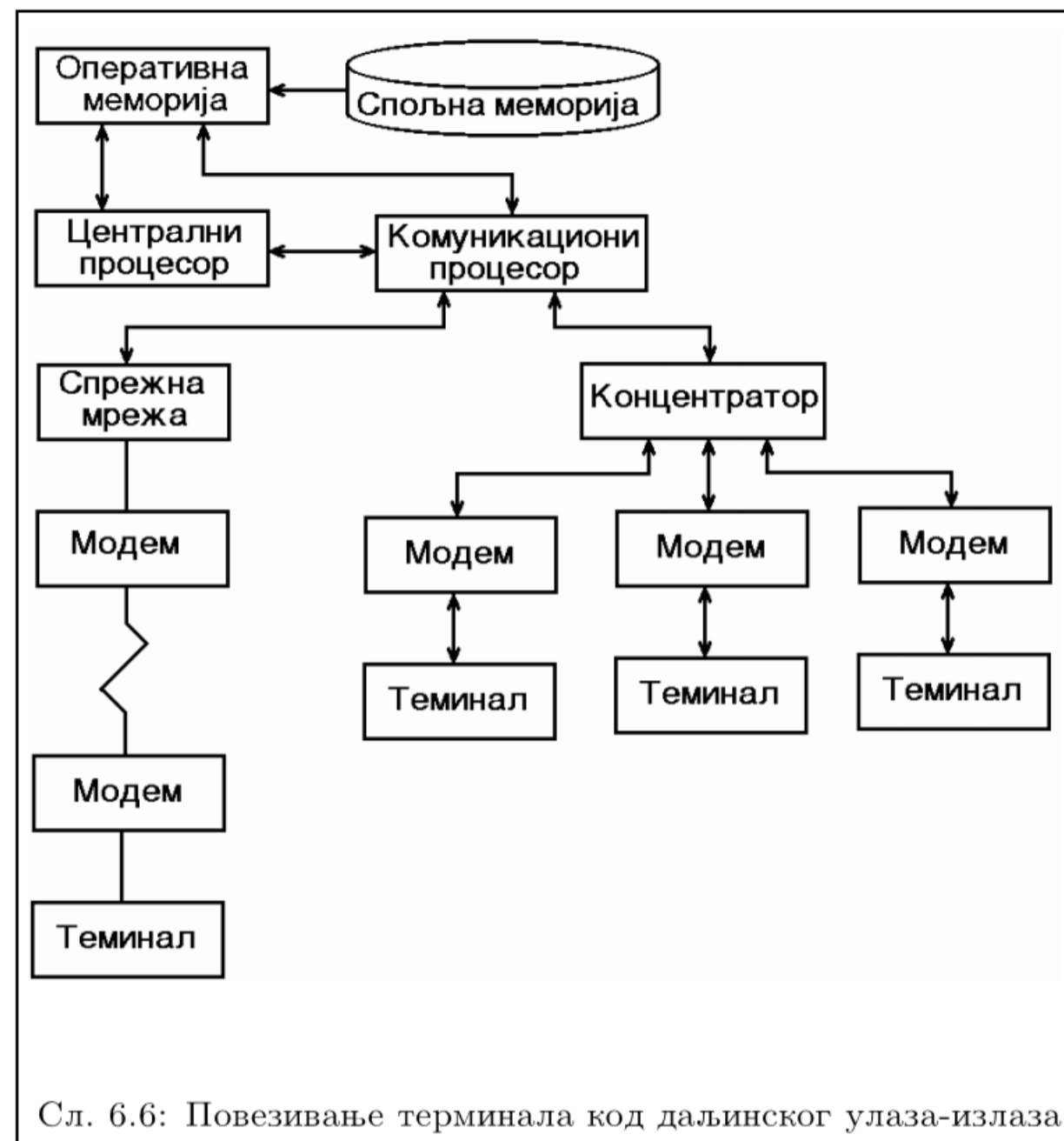
Рачунарски систем који произведе излаз за унапред задато време назива се систем за рад у реалном времену или систем реалног времена. То значи да је размак између улазног времена захтева за обрадом и излазног времена те обраде значајан и мора бити довољно мали да одзив буде правовремен. Пример рада у реалном времену чине системи за управљање технолошким процесима или системи за управљање пројектилима.

Најважније својство система за рад у реалном времену је то што он мора да буде у стању да прихвати и обради податке у времену када је нека појава у току, и да резултате обраде за модификовање неких параметара те појаве благовремено повратно саопшти. Медјутим, овде време одзива, тј. временска разлика између настајања података и повратног саопштавања резултата, зависи од примене. На пример, да би се утицало на понашање ракете у лету, одзивно време мора се кретати у оквиру микросекунди. Код појединих технолошких процеса време одзива може се мерити милисекундама.

Системима реалног времена најчешће се сматрају они системи код којих се време одзива мери милисекундама, док се системима који раде у режиму дијалога сматрају системи у којима се време одзива мери секундама.

6.2.3 Даљинска обрада

Уколико се за приступ рачунару користи даљински улаз-излаз тада се говори о даљинској обради. На процесор је преко телекомуникационих линија везан већи број удаљених терминала, персоналних рачунара или радних станица који се опслужују у вишепрограмском режиму рада. Посредством модема или концентратора са удаљених уређаја шаљу се у рачунар команде и подаци, а на основу примљених команди и података рачунар враћа излазне резултате (Слика 6.6).



Пример даљинске обраде представља систем који примењују банке. Предност таквог система је не само у могућности веома брзог преноса података насталих у филијали до рачунара већ и у чињеници да филијала има приступ подацима који се чувају у базама података рачунара чиме је обезбеђено веома брзо долажење до њих. У ствари, овде се тежи увођењу моћних централних рачунара који су у могућности да обраде податке настале на веома широком подручју. Коришћењем терминала уводи се још један принцип, познат као вишеструки приступ. То значи да више удаљених улазно-излазних пунктова има директан и практично истовремени приступ рачунару.

6.2.4 Дистрибуирана обрада

У општем случају обрада се назива дистрибуирана обрада или расподељена обрада ако се врши у систему у коме је повезно

више међусобно независних рачунара који могу да сарађују. То су тзв. дистрибуирани рачунарски системи. У сваком рачунару дистрибуираног система врши се локална обрада, а преко телекомуникационих линија међусобно се размењују подаци који су од утицаја за обраду на више рачунара. Дистрибуирана обрада заснива се на чињеници да је неке функције обраде боље обавити локално, а друге централно. Она проширује принцип даљинске обраде постављањем рачунара на удаљене тачке и њиховим повезивањем не само с централним рачунаром него и међусобно. С појавом минирачунара, радних станица и персоналних рачунара, као и локалних рачунарских мрежа, овај начин обраде постао је реално изводљив. Основна одлика дистрибуираних система састоји се у томе да различити елементи система могу бити расподељени на одвојене делове у којима је сконцентрисан строго дефинисан посебан део процедуре обраде података и самих података.

Дистрибуирана обрада подразумева постављање рачунарске опреме за локалне обраде у удаљене тачке. Ова опрема се користи за обраду на месту настанка промена. Када је то потребно, могућ је приступ централним базама података у главном рачунару или слање сумарних извештаја. Уз то, међусобним повезивањем рачунара, на пример у случају рачунара постављених у фабрикама, омогућује се координација производне активности, планирање производње, одржавање оптималних залиха и слично.

Концепт дистрибуиране обраде има више различитих варијаната. Ако се ради о јединственој конфигурацији рачунарског система, коришћење више посебних процесора понекад се сматра једном врстом дистрибуиране обраде јер сви процесори у конфигурацији паралелно (истовремено) учествују у решавању једног проблема. У том случају се каже да су процесори "чврсто спрегнути".

Други облик дистрибуиране обраде јесте коришћење процесних елемената који су физички одвојени и слабије спрегнути. Типичан пример овакве обраде је коришћење интелигентних терминала и удаљених подсистема за управљање у индустрији. Овде терминали за интелигентно прикупљање података, подсистеми за управљање процесима и други процесни елементи могу бити расподељени по територији фабрике. Одређена ограничена обрада врши се на удаљеним јединицама, док се сумарни резултати преносе главном рачунару путем преносних линија.

У општем случају обрада се сматра дистрибуираном или расподељеном ако се врши у систему у коме је повезано више међусобно независних рачунара који могу да сарађују. То су тзв. дистрибуирани рачунарски системи. У сваком рачунару дистрибуираног система врши се локална обрада, а преко локалних рачунарских мрежа или телекомуникационих линија међусобно се размењују подаци који утичу на обраде на више рачунара.

Основна одлика дистрибуираних система састоји се у томе да различити елементи система могу бити расподељени на одвојене делове у којима је сконцентрисан строго дефинисан посебан део процедуре обраде података и самих података. Концепција се одликује следећим карактеристикама дистрибуиране обраде:

- мноштво рачунарских уређаја,
- целовитост (јединство) система,
- електрична повезаност система,
- високи ниво узајамног деловања између елемената система.

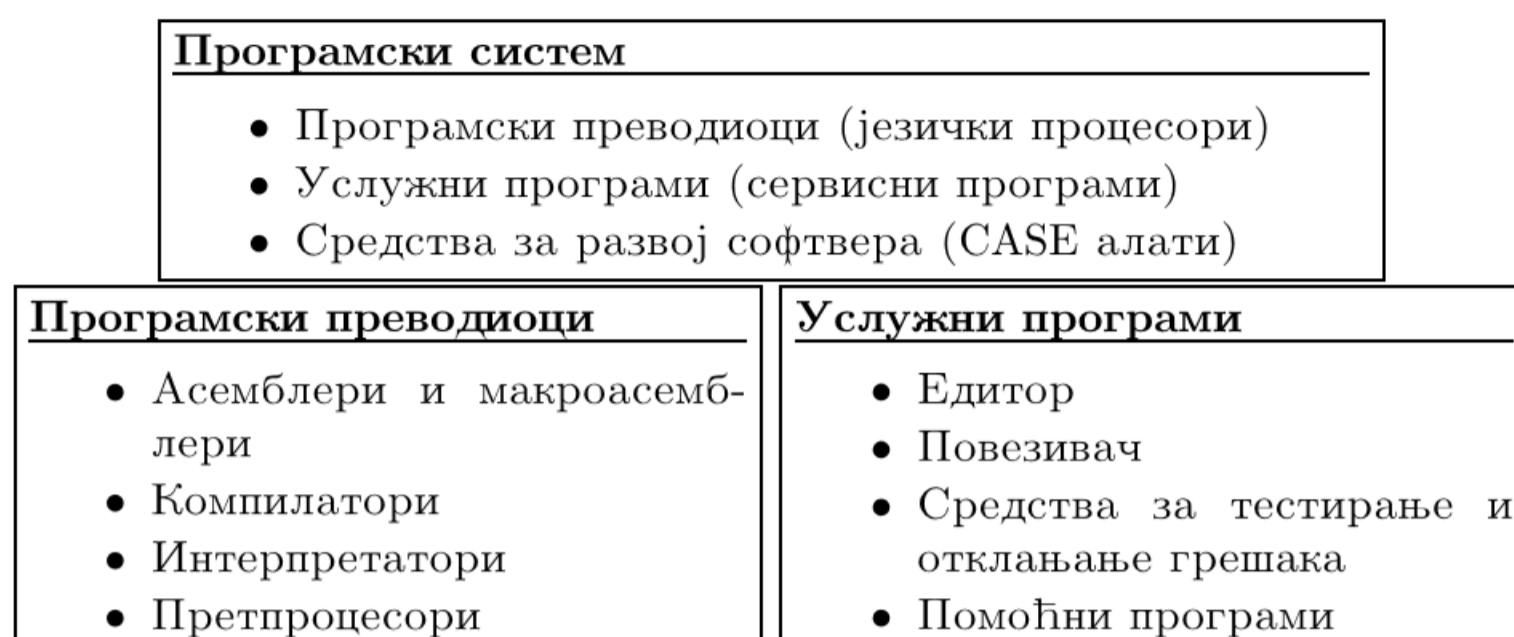
Ако је већи број терминала и рачунара повезан телекомуникационим каналима, онда они чине систем који се назива рачунарска мрежа. Обрада у рачунарској мрежи представља проширење концепта дистрибуиране обраде. Коришћење различитих конфигурација локалних рачунарских мрежа врло је погодно за дистрибуирану обраду.

6.3 Програмски систем

Програмски систем садржи програме који служе за аутоматизацију припреме и тестирања корисничких програма. То су програмски језици, програмски преводиоци (асемблери, компилатори и интерпретатори), едитори, библиотеке потпрограма и услужни програми. Основна подела програмског система је на програмске преводиоце (језичке процесоре) и услужне програме, (Слика 6.7).

6.3.1 Програмски преводиоци

Термин програмски преводиоци означава језичке процесоре који прихватају програме писане на неком од програмских језика и



Сл. 6.7: Компоненте програмског система

генеришу функционално еквивалентне програме на неком другом језику: Компиlator је преводац чији је улазни језик виши програмски језик, а излазни језик је машински језик. Интерпретатор је преводац чији је улазни језик виши програмски језик а излазни језик је нека посредна форма која се интерпретира (извршава). За разлику од асемблера компилатор и интерпретатор, по правилу, за једну наредбу написану у вишем програмском језику генеришу више инструкција машинског језика. Компиlator и интерпретатор се разликују по томе што компилатор преводи на интерни машински језик читав програм па се тек онда приступа његовом извршавању, док интерпретатор преводи једну наредбу из вишег програмског језика на низ инструкција машинског језика, приступа извршавању ових инструкција, потом прелази на превођење и извршење следеће наредбе, итд.

Компилатори

Компилација програма се може разложити на два главна задатка:

- (а) анализу - утврђивање структуре изворног програма, и
- (б) синтезу - генерисање (конструкцију) циљног програма еквивалентног изворном програму.

У процесу анализе изворни програм се конвертује у апстрактну форму која садржи сва битна својства алгоритма. Анализа обухвата лексичку анализу и син-таксну анализу. Лексичком анализом

се разлазу улазне јединице одређеног значења, као што су нпр. имена, константе, резервисане речи, оператори. Семантичка анализа, представља процес одлуђивања да ли је низ улазних симбола реченица датог језика и, ако јесте, одређивање синтаксне структуре тог низа.

Процес синтезе садржи два посебна дела: генерисање кода и асемблирање. Генерисање кода трансформише апстрактни изворни програм у еквивалентни машински програм. Асемблирањем се врши конверзија машинских наредби у одговарајући излазни формат и формира се објектни модул прилагођен обради од стране повезивача циљне машине.

Интерпретатори

Интерпретаторски начин реализације преводиоца највише се примењује код интерактивних програмских језика, као што је нпр. бејзик (енгл. BASIC) и код разних командних језика. Кад год стигне нова улазна линија даље акције преузима контролер обраде који позива:

- (а) програм хомогенизатор ради конверзије улазне програмске линије у неки стандардни интерни формат, или
- (б) синтаксни анализатор и генератор кода ради синтаксне провере и генерисања машинског кода, или
- (в) процесор команди ради обраде управљачке (контролне) наредбе, нпр. RUN, али не и њеног извршења.

Извршење преведеног програма спада у надлежност извршиоца. Реализује се читањем одговарајућег машинског кода и његовим извршавањем. Стога је стварно извршавање преведеног програма мешавина интерпретативног и машинског кода, с тим што се већина наредби извршава преко машинског кода а контрола преласка с једне на другу наредбу подржава се интерпретативно.

6.3.2 Услужни програми

Едитори

Припрема (уношење) програма обавља се у посебној фази развоја програма која се назива едитовање, а део системског софтвера који

реализује ту фазу назива се едитор програма или едитор текста. Намењен је за унос и модификацију текстуалних података, нпр. програма на вишим програмским језицима, књига и других текстуалних докумената. Резултат едитовања програма је изворни програм.

Едитори текста чине битан део корисничког интерфејса и најчешће су коришћени делови системског софтвера у интерактивним рачунарским системима. За већину корисника они представљају главну спрегу са рачунаром. Могу бити линијски едитори, када се текст посматра као низ линија раздвојених управљачким знаком за крај линије, или екрански едитори. Код екранских едитора садржај екрана представља прозор у текст унутар којег се курсор може довести на место где ће се вршити уметање, брисање и друге едиторске функције.

Једна од важних карактеристика едитора је скуп едиторских функција које су стављене на располагање кориснику. Таксономија могућности едитора обухвата потенцијално преко 200 функција које се њиме могу извршавати. Овде ћемо описати само неке од најважнијих и најчешће коришћених функција над текстуалним објектима, где се под објектом подразумева: знак, реч, линија, реченица, параграф и секција, који као целина подлежу одређеној операцији едитовања:

- уметање објеката у текст,
- уклањање (брисање) објеката из текста,
- замена објеката у тексту новим објектима,
- премештање објеката с једне на другу позицију у тексту,
- копирање објеката на разне позиције у тексту,
- раздвајање и спајање више објеката у тексту на више делова,
- селекција (означавање) појединих делова текста,
- тражење објекта у тексту, итд.

Повезивачи

Резултат превођења програма помоћу компилатора не добија се директно у машинској већ у форми која се назива објектни модул. Да би се програм могао извршавати у рачунани неопходно је довести га у извршну форму и сместити га у оперативну меморију.

Извршна форма програма се добија повезивањем објектних модула у једну целину тзв. модул пуњења и његово свођење на директну машинску форму (машински језик). Целокупан посао око генерисања извршне форме и њеног смештања у оперативну меморију обавља се у посебној фази развоја програма која се назива повезивање. Ова фаза следи после процеса превођења. Део системског софтвера који реализује функцију повезивања назива се повезивач или едитор веза. Постоје разне верзије повезивача а понекад они врше и пуњење извршне форме програма у оперативну меморију.

Програми за тестирање и отклањање грешака

Отклањање грешака или пречишћавање програма представља процес идентификације и одстрањивања локализованих грешака из програма. Насупрот томе тестирање настоји да установи да ли грешке постоје, али их не изолује нити их отклања.

Отклањање грешака у програму представља дводелни процес који почиње индикацијом о постојању грешке као резултата успешног тестирања, и наставља са следећим активностима:

- (а) прецизно одређивање природе уочене грешке и
- (б) локализација и отклањање грешке.

Помоћни програми

Помоћни програми представљају скуп програма који нису фундаменталне природе, али се њима проширују могућности рачунарског система и тиме побољшава квалитет услуга расположивих кориснику. Број и врста ових програма који су заступљени у системском софтверу се разликује од система до система. Овде ћемо навести само неколико типова помоћних програма:

- формирање програмских библиотека,
- одржавање датотека: брисање, формирање, спајање, копирање, преименовање, итд,
- коришћење датотека: модификација, ажурирање, читање, приказ на екрану, итд.
- рад са каталозима (именицима),
- конверзија датотека из једног формата у други,

- штампање садржаја датотека,
- упоређивање садржаја две датотеке, итд.

6.3.3 Средства за развој софтвера

У развоју софтвера се отишло до примене софтвера пете генерације. У петој генерацију софтвера убраја се читав скуп метода и техника аутоматизованог пројектовања интегрисаних информационих система. Једним именом их називамо, CASE (Computer Aided Software Engineering) алати.

Основне карактеристике CASE алата су у томе што дефинишу методологију и технику за аутоматизацију софтверских решења.

Компоненте CASE омогућавају:

- сакупљање, чување, и одржавање свих информација реалног система;
- дефинисање структуре и релација у бази података;
- изглед екрана и излазних извештаја;
- дефинисање програма за приступ бази података, омогућавајући њену заштиту и интегритет;
- интерактивни рад корисника, манипулације с подацима из базе, симулирање тока процеса и решења апликација;
- производњу свих оперативних компоненти аутоматски, описе базе података, техничку и корисничку документацију, датотеке са "help" и "eggo" процедурама и сл.

Уз аутоматизацију наведених решења, долази још и стандардизација и конзистентност програма, тако да CASE софтвери, не само да уносе велике новине у виду информационог инжењеринга, већ и представљају значајно повећање продуктивности и ефикасности коришћења хардверске и софтверске опреме, како од стране професионалних информатичара, тако и од стране крајњих корисника.

6.4 Оперативни систем UNIX

Ken Thompson, Dennis Ritchie и истраживачи из AT&T Bell лабораторија развили су оперативни систем Unix 1969. године,

укључивши многе могућности истраживачког пројекта MULTICS (Multiplied Information and Computing Service). Систем су обликовали за потребе истраживачког окружења, пројектујући га тако да се извршава на мини-рачунарима. Од свог почетка, Unix је био ефикасан вишекориснички оперативни систем који је у стању да обавља више задатака истовремено.

Unix систем постао је популаран у Bell лабораторијама будући да је све више и више истраживача почињало да га користи. Dennis Ritchie је сарађивао са Kenom Thompsonom 1973. године, поново написао програмски код система Unix у програмском језику C. Unix је постепено растао, од пројекта који је обликовала једна особа до стандардног софтверског производа који дистрибуирају многи продавци, као што су Novell и IBM. У почетку, Unix је сматран за истраживачки пројект. Прве верзије Unixа дистрибуиране су бесплатно катедрама за рачунарство многих познатих универзитета. Током седамдесетих, Bell лабораторије почеле су да издају званичне верзије Unixа и да издају дозволе за системе различитим корисницима. Један од ових корисника била је катедра Computer Science калифорнијског универзитета Berkeley. На Berkeleyу су систему додане многе нове могућности које су касније постале стандардне. Berkeley-ы је 1975. године издао сопствену верзију Unixа, познату под именом Berkeley Software Distribution (BSD). Ова BSD верзија Unixа постала је главни такмац верзији AT&T Bell лабораторија. У AT&T-у развили су неколико истраживачких верзија Unixа, а 1983. године је AT&T издао прву комерцијалну верзију, названу System 3. За њом је касније следио System V, верзија која је постала подржани комерцијални софтверски производ.

У исто време, BSD верзија Unixа развијала се кроз неколико издања. Касних седамдесетих, BSD Unix је постао основа истраживачког пројекта који је спровела владина агенција DARPA (Department of Defense's Advanced Research Projects Agency). Резултат је био тај да је 1983. године Berkeley издао моћну верзију Unixа названу BSD издање 4.2. Ово издање се одликовало софистицираним управљањем датотекама као и могућностима умрежавања заснованим на Интернет мрезним протоколима - истим протоколима који се сада користе за Интернет. BSD издање 4.2 било је широко дистрибуирано и усвојили су га многи произвођачи, као што је Sun Microsystems.

Средином осамдесетих издвојила су се два конкурентска стан-

дарда, један заснован на AT&T верзији Unixа и други заснован на BSD верзији. AT&T-ове лабораторије Unix System Laboratories развиле су System V издање 4. Неколико других компанија, попут IBM-а и Hewlett-Packarda, основале су организацију OSF (Open Software Foundation) како би развиле сопствену стандарду верзију Unixа.

Развојем изворне верзије оперативног система, појављивале су се и нове верзије различитих произвођача. Међутим, за све верзије Unix-а било је заједничко да је то модуларни, вишепрограмски оперативни систем који омогућава рад са поделом времена. Са друге стране, Unix је независан од хардвера, погодан за развој програма и лак за коришћење.

Unix има слојевиту структуру класичних оперативних система, а њу формирају:

- централни део
- језгро,
- помоћни системски програми - алати и
- програмски преводиоци и кориснички програми.

Централни део оперативног система се назива језгро или нуклеус (енгл. kernel). Овај део оперативног система је задужен за директну комуникацију са хардвером. Управљање хардвером Unix остварује преко специјалне датотеке (SPECIAL FILE). За комуникацију језгра са горњим слојевима користе се тзв. системски позиви.

Основне компоненте језгра су:

- систем за управљање задацима,
- систем за вођење евиденције о датотекама и
- систем за управљање улазом-излазом.

6.4.1 Управљање задацима

Систем за управљање задацима остварује:

- комуникацију између задатака,
- синхронизацију између задатака,

- планирање задатака,
- управљање меморијом.

Задатак или процес у Unix-у представља програм који се извршава. Он може да се нађе у два основна режима рада:

- у системском режиму (енгл. *system state*) - повлашћеном режиму или режиму језгра, када задатак извршава језгро и када приступа системским сегментима података;
- у режиму корисника (енгл. *user state*), када задатак изводи корисничке програме и приступа корисничким сегментима података.

Прелазак из режима корисника у системски режим остварује се преко системског позива, док се прелазак из системског режима у режим корисника дешава када је системска услуга (приступање системским сегментима података) - извршена, или када се, ако услуга није могла бити извршена, пошаље одговарајућа порука о грешкама.

Креирање задатака се остварује посебним системским позивом `fork`. Када почне учитавање програма језгра са диска, активира се задатак "0" или језгро. Све друге новије процесе креира неки од постојећих задатака. Сваки задатак добија свој идентификациони број. Тако, на пример, језгро има идентификациони број 0; следећи задатак, који надгледа креирање осталих задатака (`init` процес), креира се од стране језгра, системским позивом `fork` и он добија идентификациони број 1. Остали задаци могу имати различите идентификационе бројеве.

Комуникација између задатака и њихова међусобна синхронизација може се остварити на више начина:

- повезивањем задатака помоћу цеви (`pipe`) проток бајтова између процеса у једном смеру,
- повезивањем задатака преко порука,
- повезивањем задатака преко заједничке меморије,
- повезивањем задатака преко семафора.

Планирање задатака представља доделу процесорског времена задацима. Unix користи алгоритам кружног планирања са редовима у више нивоа (RR-алгоритам - *round robin with multilevel feedback*). Кориснички задаци добијају динамичке приоритете, зависно

од количине већ добијеног процесорског времена. Задаци који су добили више процесорског времена, добијају ниже приоритете и обратно.

6.4.2 Управљање меморијом

Управљање меморијом се остварује креирањем виртуелног адресног простора. Меморија се задацима додељује у страницама. Сваки задатак добија слику меморије која означава садржај који је у меморији за време извођења задатка. Слика меморије додељене одређеном задатку може се пребацивати из оперативне меморије на диск (енгл. *swapping*), ако неки од задатака вишег приоритета затражи оперативну меморију.

6.4.3 Управљање датотекама

Систем за управљање датотекама обавља следеће послове:

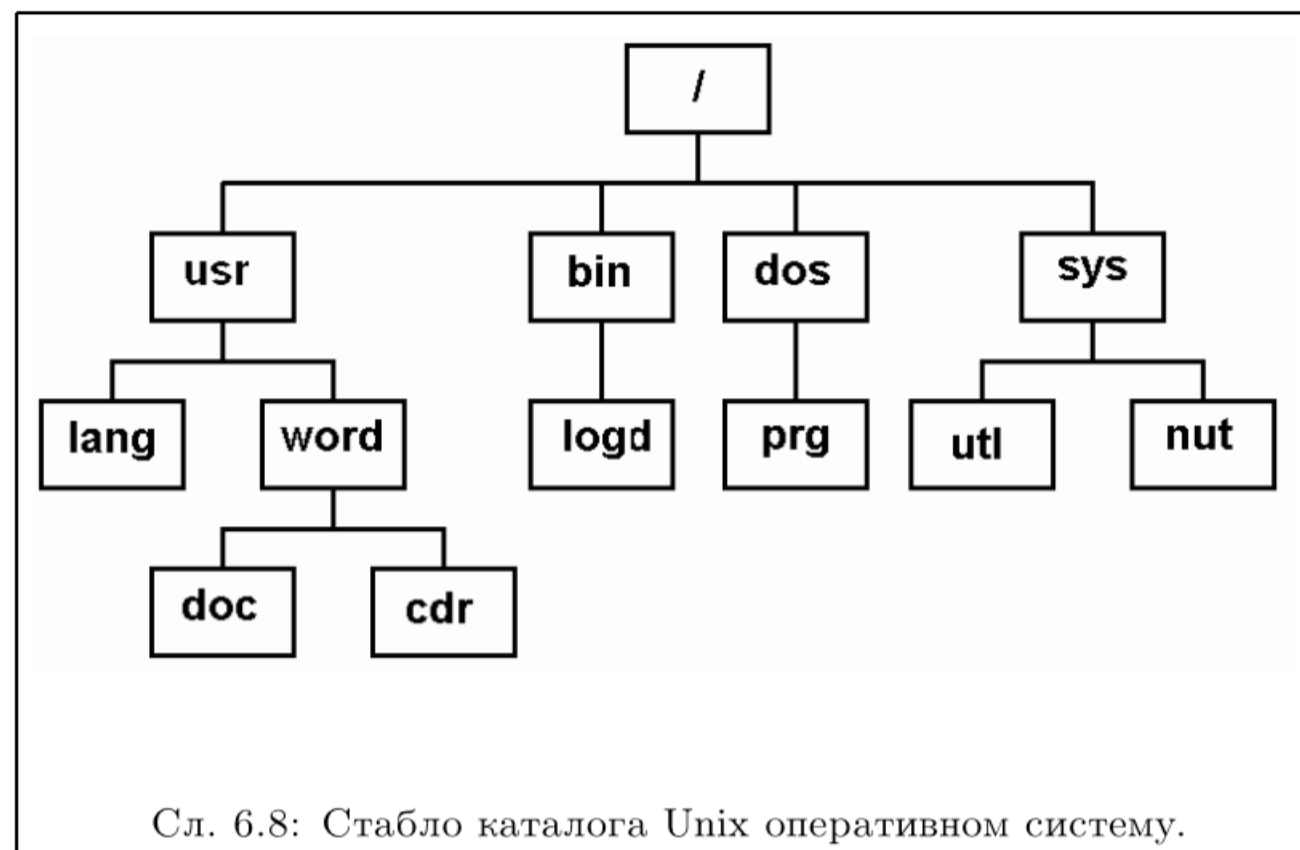
- додељује простор на спољној меморији (дискони, магнетне траке),
- води евиденцију о слободном простору,
- проналази податке (претражује садржај датотека) по захтеву корисника,
- води рачуна о приступу датотекама.

Датотеке су организоване на хијерархијски начин формирајући структуру стабла каталога (директоријума) и датотека. Пример организације система датотека у Unix-у је дат на слици 6.8.

Кориснички програми и подаци су у Unix оперативном систему смештени у виду неколико типова датотека. Постоји више типова датотека:

- обичне датотеке,
- датотеке каталози (именици, адресари или директоријуми),
- FIFO датотеке
- специјалне датотеке.

Типови датотека дефинишу њену функцију и начин приступа.



Обичне датотеке садрже податке организоване на одређени начин. Не постоји ограничење на формат обичних датотека. Обичне датотеке могу бити бинарни програми, програми записани у неком изворном програмском језику или подаци у различитим облицима.

Датотеке каталози су чворови који повезују структуру датотека у целину. Ове датотеке имају већ унапред дефинисани формат записа и третирају се као и остале датотеке, осим што поседују додатну заштиту.

FIFO датотеке омогућавају комуникацију између задатака који нису међусобно повезани. Најчешће се FIFO датотеке користе када је потребно да се оствари комуникација више задатака са једним одређеним задатком.

Специјалне датотеке описују физичке уређаје. Податак уписан у специјалну датотеку прослеђује се одговарајућој периферној јединици и обрнуто: када се подаци читају са периферног уређаја, предају се задатку који је тражио податке. У ствари, специјалне датотеке представљају само показиваче на адресе програма који се налазе у самом језгру оперативног система и који управљају радом периферних јединица.

Обичне, FIFO и специјалне датотеке у структури стабла датотека могу се представити као листови, док датотеке каталози (именици) представљају гране стабла.

6.4.4 Управљање улазам-излазом

Управљање улазом-излазом се остварује преко улазно-излазног система. Unix систем је оријентисан на појединачни улаз-излаз који се састоји од бафер-меморије и везних управљачких програма (драјвера).

Улазни низ представља низ знакова који се завршавају неком од ознака:

- ознаком за крај линије (RETURN),
- ознаком за крај датотеке (ASCII EOF),
- ознаком за брисање знака (ASCII DEL).

Сваки од ових знакова проузрокује прекид пријема улазног низа знакова и иницира обраду унетих података. Дужина улазног низа, очигледно, није унапред дефинисана. На тај начин се остварује већа независност периферних уређаја од осталих делова система и флексибилност у раду са подацима, али је потребно да за сваки тип уређаја постоји одговарајући везни управљачки програм. Разликују се три типа периферних уређаја, па самим тим и управљачких програма за њих:

- знаковно оријентисани управљачки програми: за штампаче, терминале, монитор итд.;
- блок оријентисани управљачки програми: за дискове и магнетне траке;
- управљачки програми за мрежне интерфејсе.

Програми у Unix оперативном систему све спољње уређаје виде као датотеке.. Аналогно томе, улазно-излазни систем поседује четири основна позива:

1. open - отварање датотеке,
2. close - затварање датотеке,
3. write - писање у датотеку и
4. seek - претраживање датотеке.

6.4.5 Помоћни системски програми алати

Око језгра оперативног система Unix, налазе се помоћни системски програми. Најзначајнији међу њима су:

- shell, sh - интерпретатори командног језика;
- ed, vi, ex - едитори текста;
- ld - пунилац (енгл. loader);
- as - асемблер.

Најчешће се комуникација корисника са рачунаром остварује преко команди које интерпретира шелл. Према овом програму се Unix односи као према сваком другом корисничком програму, што значи да shell није резидентан у меморији рачунара (као на пример језгро), већ се по потреби учитава. То значи да shell представља један од "обичних" програма који сваки корисник може мењати и прилагођавати себи, а то опет значи да поред стандардних команди корисник може додавати своје. Командна линија садржи:

- име команде,
- аргументе које команда захтева (они се одвајају бланко знацима).

Shell кориснички интерфејс "препознаје":

- извршавање програма у позадини,
- проток података између команди (pipe),
- редирекцију (преусмеравање) команди излазне вредности команде се могу слати у неки одређени фајл или неки од прикључених уређаја.

У табели 6.1 су дате неке од команди UNIX-а.

Када се нека команда или датотека (програм) покрене на извршење, shell му додељује три нове датотеке:

- за улаз,
- за излаз,
- за грешке.

Табела 6.1: Неке команде UNIX оперативног система

Команда	Значење
cd	Промена текућег каталога
chmod	Дефинисање начина приступа датотеци
chown	Промена власника датотеке
cmp	Упоредивање садржаја две датотеке
cp	Копирање датотека
date	Приказ или мењање текућег датума
find	Претраживање стабла каталога
kill	Затварање тренутно отвореног задатка
ls	Приказивање садржаја текућег каталога
mail	Слање или пријем електронске поште
mkdir	Креирање новог каталога
passwd	Промена лозинке
pwd	Приказ текућег каталога
rm	Брисање задате датотеке
rmdir	Брисање текућег каталога
stty	Приказ или промена опција терминала
who	Испис листе тренутних корисника

Оваква концепција омогућава да се излаз из једне команде искористи као улаз у другу команду. Такав начин рада се назива *pipe* (цев).

Новије верзије Unix оперативног система имају графички кориснички интерфејс, што омогућава почетницима лакше упознавање и коришћење. Графичке радне станице су обично прикључене на локалну рачунарску мрежу и, мада углавном поседују сопствене дискове и друге важније ресурсе, ослањају се на један већи рачунар - сервер за датотеке. Стандардни систем за управљање прозорима у Unix-у, под називом XWindow омогућава извршавање САД програма и других апликација које захтевају додатне графичке могућности. Наравно, испод слоја различитих апликативних графичких програма налази се иста - Unix основа.

6.5 Оперативни систем Windows

Оперативни систем Windows (енгл. window - прозор) представља производ фирме Microsoft. Намењен је различитим рачунарским системима: од стоног РС рачунара, преко локалних и глобалних рачунарских мрежа, до суперсервера вишепроцесорских рачунара

базираних на RISC процесорима.

MS Windows се као оперативни систем појавио осамдесетих година. Он је представљао платформу под којом су могли да се извршавају многобројни програми писани за MS DOS, али и да омогући:

- унифициран графички кориснички интерфејс,
- потпуно коришћење могућности савремених процесора,
- ефикасније коришћење оперативне меморије,
- динамичку размену података,
- виртуелни меморијски систем,
- вишепрограмски рад,
- подршку за мултимедију итд.

Популарност овог оперативног система почиње од верзије Windows 3.0. Касније верзије: Windows 95, Windows 98 и Windows XP још више су учврстиле позиције овог оперативног система. Данас је Windows један од најпродаванијих оперативних система, пре свега за персоналне рачунаре. За различите типове рачунара постоје различите верзије Windows-а:

1. Windows 3.1 је намењеним стоним PC рачунарима класе 386 и 486 и Pentium PC, као и за разне преносне рачунаре;
2. Windows for Workgroups (прозори за радне групе) или Windows 3.11, представља у ствари Windows 3.1, са подршком за локалне мреже и интегрисану електронску пошту;
3. Windows 95 је наследник Windows-а 3.1 x, представља 32-битну верзију оперативног система намењену рачунарима са нешто бољим карактеристикама и са већим меморијским захтевима. Ова верзија представља комплетан оперативни систем који се дефинитивно одваја од MS DOS-а, али га и даље подржава; Windows NT је оперативни систем за снажне радне станице и мрежне сервере.

Програми за пружање услуга оперативног система Windows су садржани у три модула:

- модул за општу подршку
- језгро (енгл. kernel),
- графички модул (GDI),

- кориснички модул - модул за управљање прозорима.

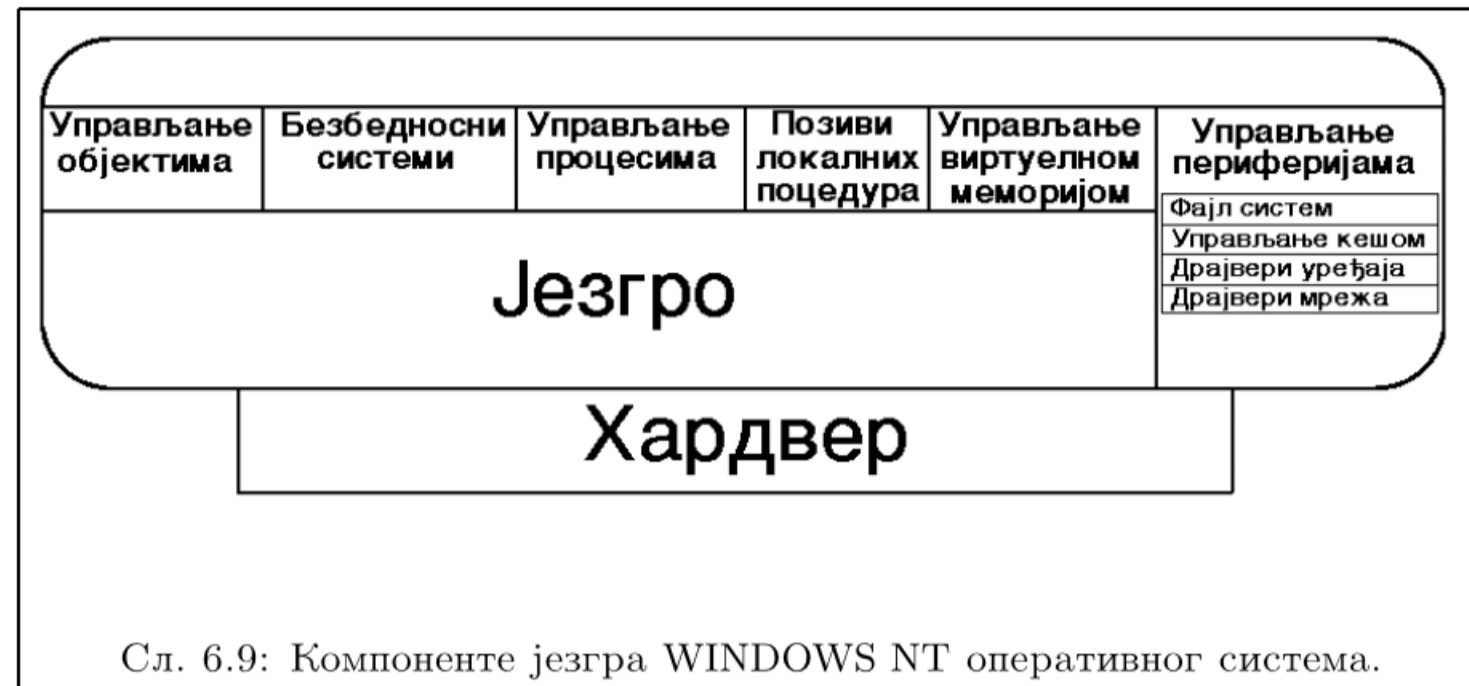
Windows 95 се ослања на овакву структуру, али садржи комплетнију подршку за рад са 32-битним програмима, омогућава већу дужину имена датотека, другачији, ефикаснији приступ, оптичким дисковима и осталим периферним уређајима и додатне могућности за рад у мрежи. Ипак, највеће могућности пружа верзија Windows NT.

Windows NT (енгл. NT - New Tehnology нова технологија), представља 32-битни оперативни систем креиран са амбицијом да у будућности буде основа за све РС рачунаре, радне станице и сервере. Обједињује све добре особине које би требало да поседује савремени оперативни систем. Поред особина претходних верзија, овај оперативни систем садржи:

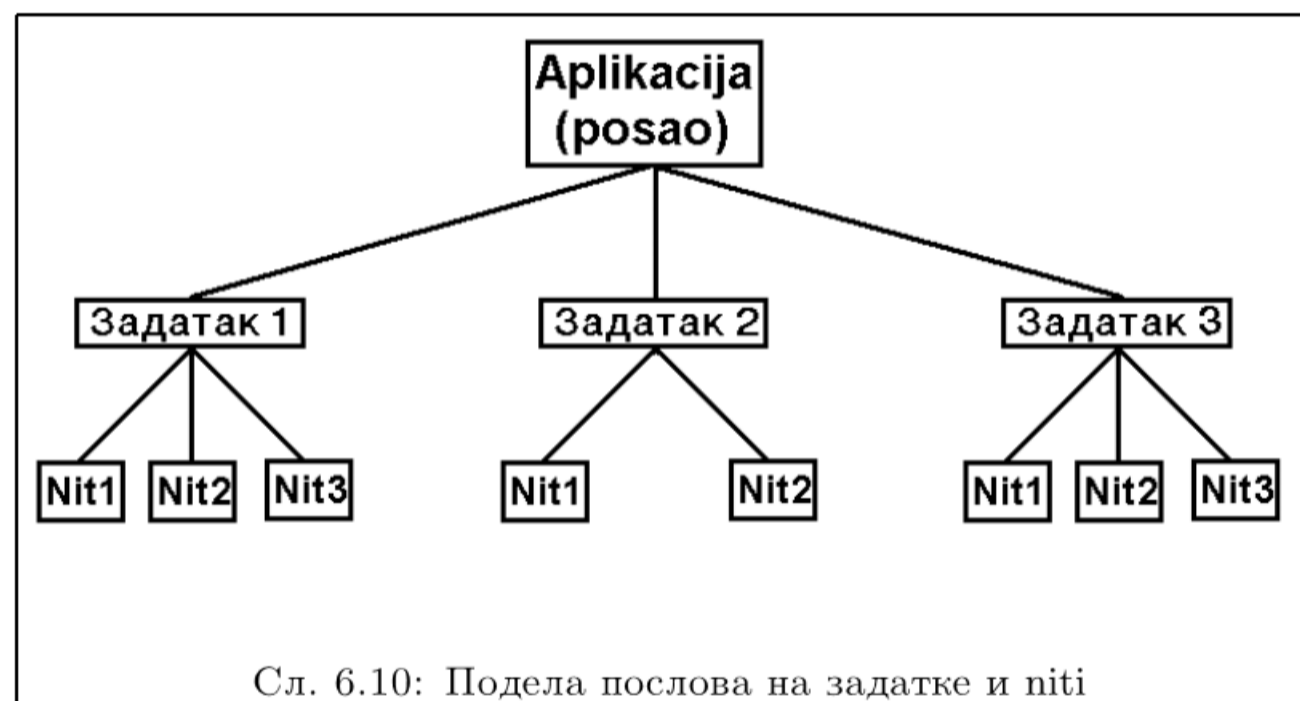
- (а) могућност рада на различитим хардверским платформама (чак и на рачунарима са различитим типовима централних процесора);
- (б) подршку свим типовима рачунарских мрежа и унапређене могућности за дистрибуирану обраду;
- (в) могућност коришћења свих досадашњих DOS и Windows апликација;
- (г) веза са Unix апликацијама преко стандарда који дефинише потребне функције и интерфејсе: POSIX (Portable Operating System Interface based on UNIX);
- (д) компатибилност са савременим оперативним системом OS/2, производом фирме IBM;
- (ђ) високу стабилност система: грешке у појединим апликацијама не могу да проузрокују пад целог система;
- (е) подржава рад у вишепроцесним системима.

Као у оперативном систему Unix, и у верзијама Windows 3.1x, структура Windows NT је слојевита: језгро, управљање графиком и кориснички ниво. Функцију језгра има део система под називом NT Executive. Компоненте језгра су приказане на слици 6.9.

Програми под Windows NT могу да се извршавају у два основна режима: - режим језгра - кад се извршава неки системски програм, - кориснички режим - кад се извршавају апликације корисника.



Оваква организација рада је слична као и у Unix оперативном систему. Посао који треба да се обави дели се на задатке или процесе. На тај начин се остварује паралелно извршавање више делова програма или више различитих програма. Задаци се деле на нити (thread). Нит такође представља задатак у оквиру програма који се извршава паралелно са свим осталим задацима у расподељеном времену (сл. 6.10).



Разлика између задатка (процеса) и нити састоји се у следећем: оперативни систем забрањује да један задатак неконтролисано улази у податке другог задатка, већ се размена врши искључиво преко оперативног система, док је међу нитима истог задатка могућа интерна размена. Нити су сегменти једног посла и они деле податке и остале ресурсе. Због тога се независне функције

програма, које захтевају одвојене ресурсе, могу реализовати само као посебан задатак (процес). Сваком задатку посебно се додељује виртуални адресни простор, једна или више нити и остали ресурси по захтеву.

Програми у режиму језгра имају улогу да:

- прерасподељују нити задатака,
- организују комуникацију између нити,
- управљају виртуелном меморијом,
- управљају објектима,
- обезбеђују заштиту система,
- надзиру рад других управљачких програма.

У корисничком режиму се обавља остали део функција оперативног система као што је организација система датотека, формирање веза према различитим апликацијама, подршка за мрежу итд.

6.5.1 Услужни софтвер

Услужни програм је било који програм који је специфичан за одређену примену рачунара. Насупрот томе системски програми су од суштинског значаја за ефикасно коришћење рачунарског система. Услугне програме могу да развијају корисници рачунарског система за решавање својих проблема. Због тога се ови програми често називају и кориснички програми.

Напоменимо још да се често у оквиру апликационих програма могу по потреби да проширују и одређене функције оперативног система.

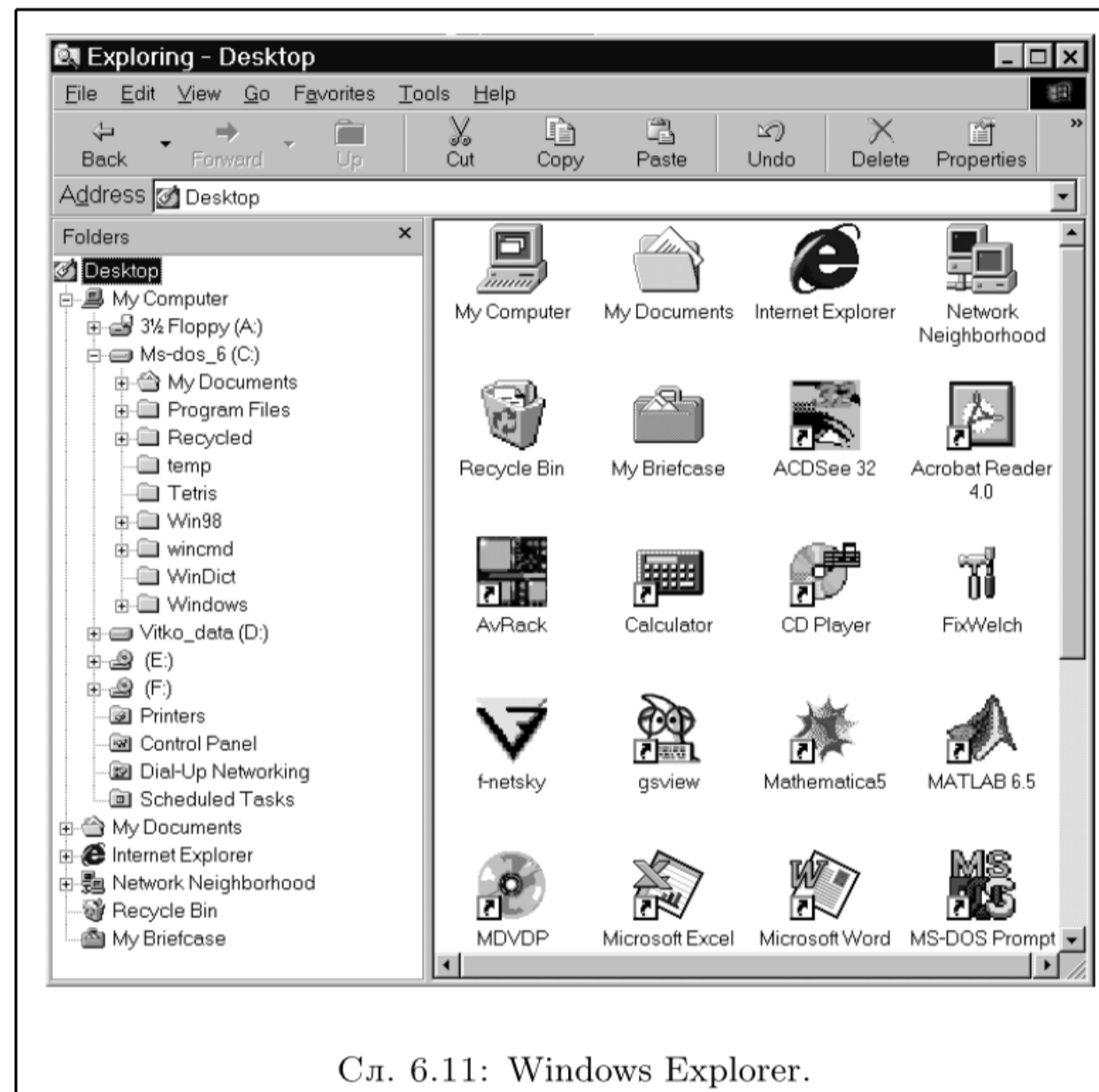
Апликациони програми се понекад деле у следеће врсте:

- појединачни апликациони програми,
- програмски пакети,
- системи програма.

Програмски пакет састоји се од више програма и представља скуп процедура за решавање проблема. Систем програма представља скуп програма чије су компоненте уско повезане.

Windows Explorer

Windows Explorer, или истраживач оперативног система је један од значајних програма у Windows оперативном систему, а функција му је управљање директоријумима или датотекама. Прозор програма је подељен на два дела као што је приказано на Слици 6.11.



Сл. 6.11: Windows Explorer.

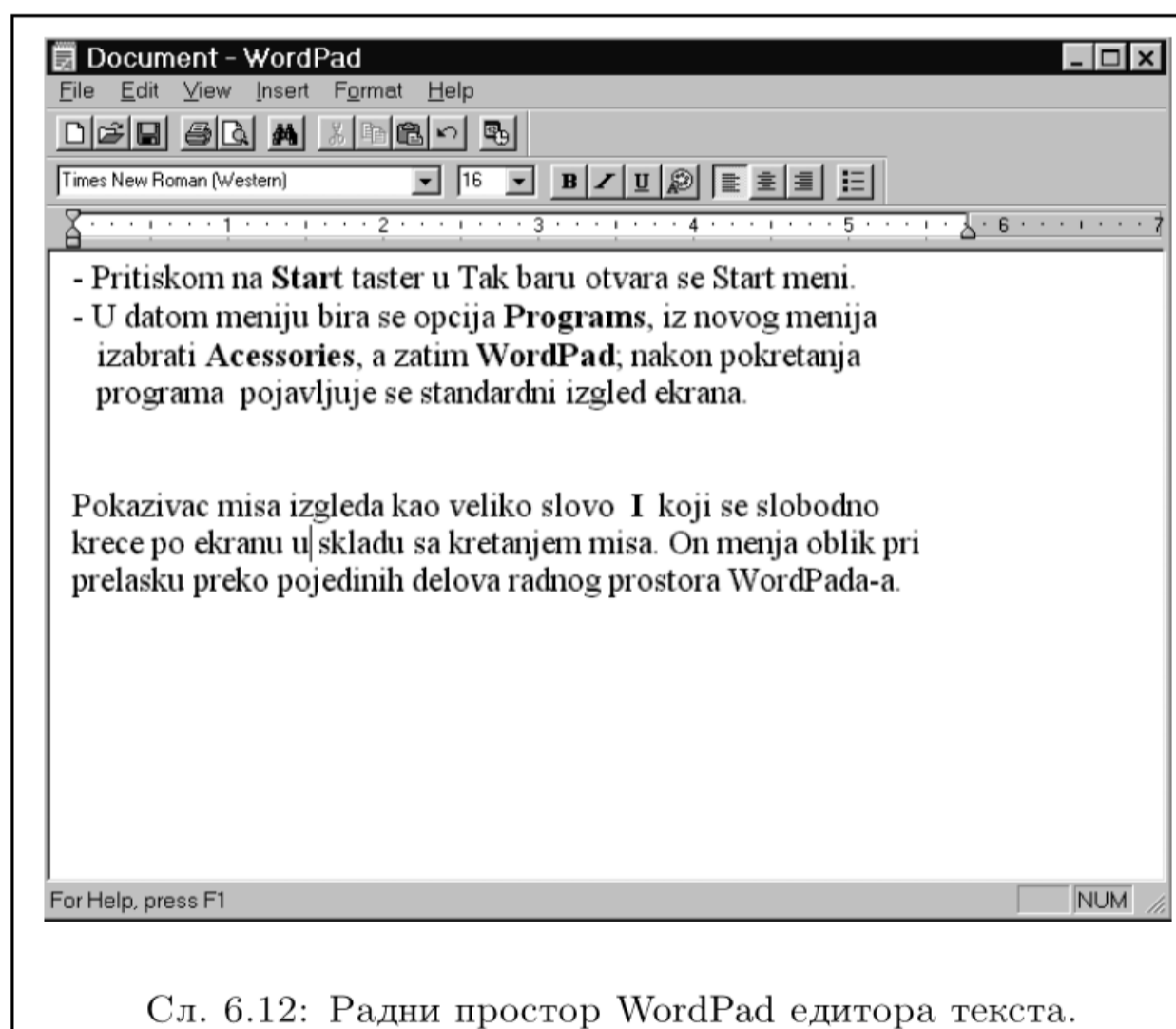
Леви панел садржи стабло директоријума, тј. структуру организације фајлова и података на меморијском систему рачунара. Десни панел садржи директоријуме и датотеке на диску изабраних на стаблу директоријума.

Испод насловне линије која садржи икону за умањивање прозора, икону за увећавање прозора и икону за затварање прозора, налази се линија са падајућим менијима који садрже команде за управљање датотекама. На пример падајући мени **E**dit омогућава следећ операције над датотекама: отварање нових, брисање старих (премештање у корпу за отпатке), промену имена, копирање итд. О нинтерфејсу са менијима бић речи касније.

Копирање и премештање датотека се може реализовати превлачењем датотеке, притиском на леви тастер миша, на неку другу датотеку или диск. Ако се датотека “превуче” на другу локацију она ће бити пренесена али не и копрана. Премештање подразумева копирање на нову локацију и брисање полазне датотеке на постојећој локацији. Копирање датотеке треба нагласити тако сто се током превлачења држи притиснут тастер < *Ctrl* >. Датотека која секопира приликом превлачења има исписани знак + поред себе.

Едитор текста WordPad

Кориснички програм WordPad представља је едитор текста који омогућује кориснику да изврши форматирање текста: избор фонта, подешавање маргина, проред између редова и слично. Стандардно се испоручује са Windows оперативним системом. Покретање WordPad-а се врши притиском на <Start> тастер у Task бару, а затим се бирају опције **Programs**, **Accessories** и на крају **WordPad**. Након покретања програма појављује се изглед екрана приказан на Слици 6.12



Сл. 6.12: Радни простор WordPad едитора текста.

Показивач миша изгледа као велико слово I који се слободно

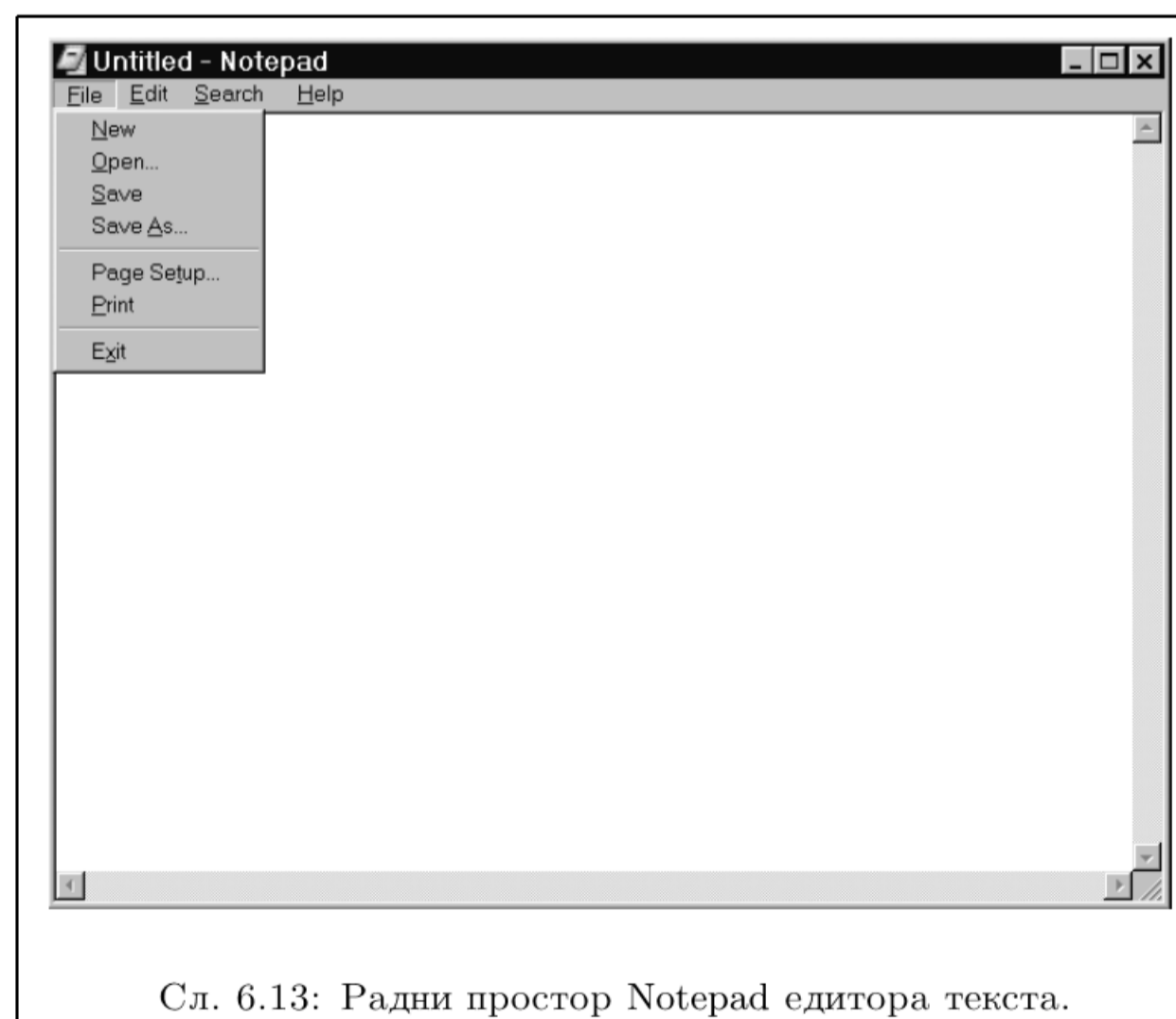
креће по екрану и складу са кретањем миша. Он мења облик при преласку преко појединих делова радног простора WordPad-а.

Хоризонтални лењир је један од елемената радног простора WordPad-а. Омогућава брзу промену маргина и увлака. Лењир се може уклонити отварањем <View> у главном менију и уклањањем знака ✓ поред речи Ruler.

Притиском на функцијску типку <F1> на екрану се појављује упуство за рад са текст едитором WordPad.

Бележница-Notepad

Текст едитор Notepad служи за унос и обрад једноставног текста. Текстови који се обрађују овим текст едитором могу да садрже само знаке који припадају ASCII кодном распореду, без икаквих додатака. То значи да у Notepadу-у није могуће поједине делове текста укосити, подебљати, писати различитом величином и бојом, поравњавати и слично. Notepad се покреће притиском на на <Start> тастер у Task бару, а затим се бирају опције **P**rograms , **A**ccessories и на крају **N**otepad, као што је приказано на Слици 6.13.



Сл. 6.13: Радни простор Notepad едитора текста.

Прозор носи назив `Untitled - Notepad`, што значи да тексту који се обрађује није дато име, и резервсан је за унос текста. Курзор је у облику усправне црте и означава место на којем ће се појавити текст који се уноси.

Притиском на функцијску типку `<F1>` на екрану се појављује упуство за рад са текст едитором `WordPad`.

Оперативни систем `Windows` садржи меморијски део, који носи назив клипборд (`Clipboard`), у који могу да се копирају различити подаци (текст, слика, фајл исл.) из било ког програма који ради под `Windows` оперативним системом. Такође, подаци из клипборда могу се укључити у сваки програм који ради под `Windows` оперативним системом. У клипборд може да се чува само један податак, а копирањем података у клипборд, предходни податак се брише.

Клипборд се највише користи при раду са бележницом као и за премештање података из једног програма у други. Команде из `Edit` менија могуће је користити употребом комбинације типки:

- копирање података у клипборд `<Strl> + <C>`,
- копирање садржаја клипборда на жељено место `<Strl> + <V>`
- премештање подата у клипборд `<Strl> + <X>`.

Наведене скраћенице су исте у свим програмима који раде под `Windows` оперативним системом и лакше их је употребљавати него бирати одговарајуће комаде у `Edit` падајућем менију.

6.6 Кориснички интерфејс

Кориснички интерфејс рачунара често је критеријум којим се оцењује лакоћа рада са рачунаром. Интерфејс који је тежак за коришћење, у најбољем случају, имаће као резултат високи ниво грешака корисника.

Кориснички интерфејс са рачунаром треба да реши на задовољавајући начин следећа два проблема:

- како може информација од корисника да буде обезбеђена рачунам,
- како може информација из рачунара бити приказана кориснику.

Почетком 80-тих година стандардни уређај за интерфејс био је ”нем” алфанумерички терминал са тастауром и зеленим или плавим знацима на црној позадини екрана. Кориснички интерфејси били су текстуални или засновани на формуларима (обрасцима, енгл. form-based). Мада ће интерфејси засновани на тексту остати у употреби још много година, корисници све више очекују апликационе системе који имају неку врсту графичког интерфејса.

Предности графичког корисничког интерфејса су следеће:

- Он се релативно лако учи и користи. Корисници без практичног рада на рачунару могу да користе интерфејс после кратке тренинг сесије;
- Корисници имају више прозора за интеракцију са системом. Прелаз од једног задатка на други могућ је без губљења увида на информацију генерисану у току првог задатка;
- Брза интеракција на целом екрану могућа је са непосредним приступом било где на екрану.

У табели 6.2 дате су најважније карактеристике графичког корисничког интерфејса.

Табела 6.2: Карактеристике графичког корисничког интерфејса.

Карактеристика	Опис
Прозори	Више прозора омогућава истовремено приказивање на екрану корисника различитих информација
Иконе	Иконама се представљају различити типови информација. На неким системима иконе представљају датотеке, на другим, иконе представљају процесе.
Менији	Команде се селекују из менија уместо да се уносе у командном језику.
Показивачки уређаји	Показивачки уређаји, као што је миш, користе се за селекцију варијанти из менија или за индикацију на екрану елемената који су од интереса.
Графика	Графички елементи могу се комбиновати са текстуалним на истом дисплеју.

6.6.1 Карактеристике корисничког интерфејса

Кориснички интерфејс мора узети у обзир физичка и ментална ограничења људи који користе рачунарске системе. Најважнија

је, вероватно, потреба да се препознају ограничења обима краткорочне меморије људи и да се избегне пренатрпаност корисника информацијама. Конзистентност интерфејса означава да системске команде и менији треба да имају исти формат, параметри треба да се преносе командама на исти начин, а интерпункција команди треба да буде слична.

Конзистентни интерфејси редукују време учења корисника. Знање стечено у једној команди или апликацији применљиво је на друге делове система. Важна је такође конзистентност интерфејса кроз подсистеме. Колико год је могуће команде са сличним значењем у различитим подсистемима треба да се изражавају на исти начин.

Корисници неизбежно праве грешке при коришћењу рачунарског система. Интерфејс може минимизирати те грешке (на пример, коришћење менија значи да ће бити избегнуте грешке које настају при куцању) али грешке се никада не могу потпуно елиминисати. Интерфејс треба да садржи средства која омогућавају да корисници изврше опоравак после својих погрешака. Ово се може остварити на један од следећа два начина:

1. потврђивање деструктивних акција. Ако корисник зада акцију која је потенцијално деструктивна, он или она ће бити упитан да потврди да је то стварно оно што је намеравао, пре него што било каква информација буде уништена;
2. обезбеђење undo средстава. Команда undo (поништити) враћа систем у стање пре него што је акција извршена. Више нивоа ундо команде су врло корисни јер корисници увек не препознају одмах да је учињена грешка. У пракси је ово скупо за реализацију. Зато већина система само омогућава да последња команда буде поништена.

На крају, интерфејс треба да има уграђену помоћ кориснику или help средства. То треба да буде интегрисано са оперативним системом и треба да обезбеди различите нивое помоћи и савета. Нивои треба да се састоје од основних информација како почети рад са системом, до потпуног описа могућности система. Средства за помоћ треба да буду структурирана, корисник не сме бити ”прегажен” информацијама када он тражи помоћ.

6.6.2 Директна манипулација

Модификовањем информација приказаних на њиховом екрану и издају команде користећи меније.

Корисници интерагују са приказаним информацијама кроз директне акције као што су замена информације, премештање информације итд. Експлицитне команде за модификацију информација нису неопходне. Модификације на приказаном моделу непосредно мењају основну информацију.

Чест пример интерфејса са директном манипулацијом јесте интерфејс који обезбеђује већина процесора речи или екранских едитора. У том случају информациони простор се састоји од низа параграфа (одељака) који су приказани кориснику као документ. За уметање текста курсор се поставља на одговарајуће место на екрану и уноси се (куца се) текст. Текст се непосредно појављује показујући кориснику последице његових акција.

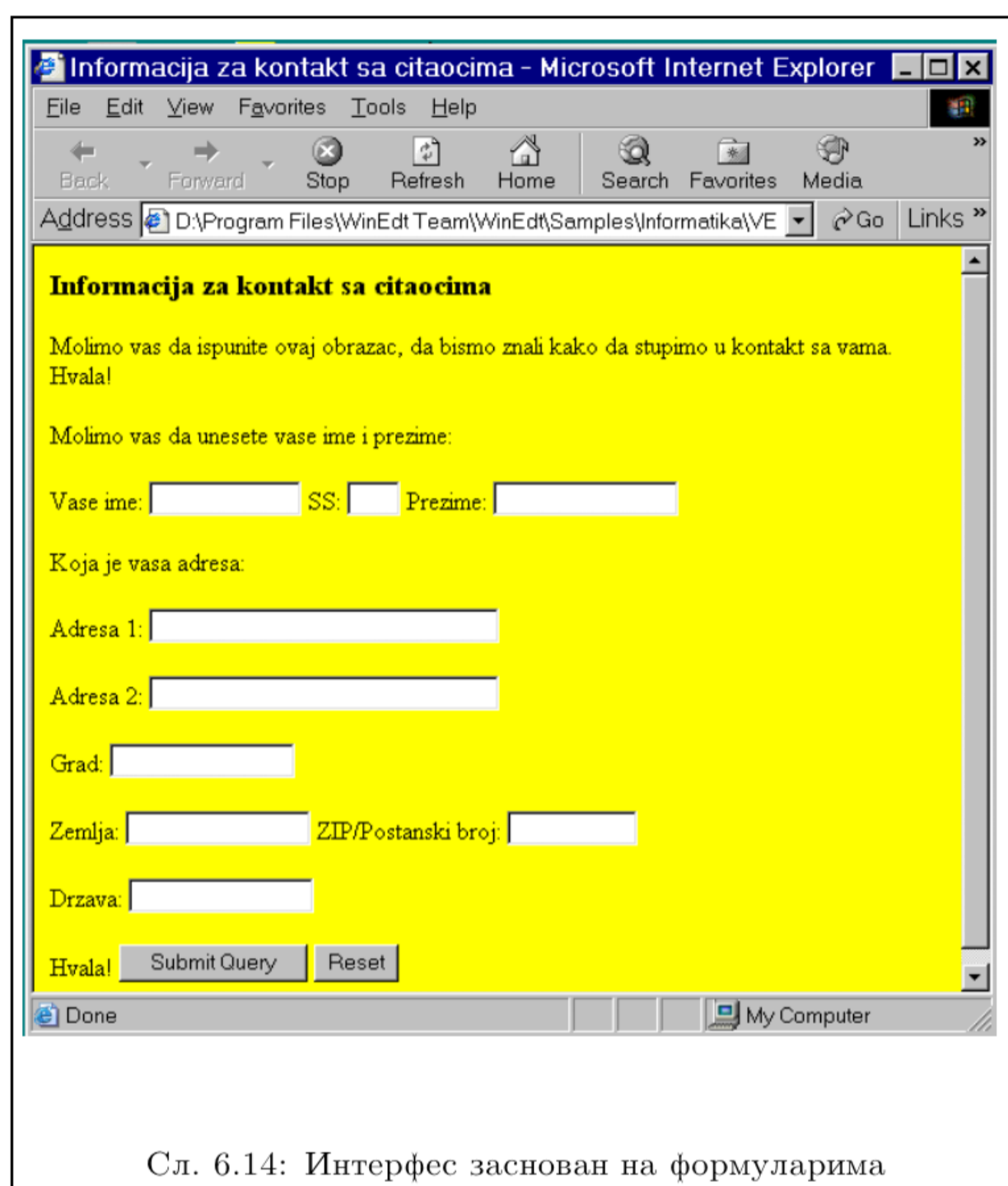
Други пример је графички кориснички интерфејс, где се кориснику приказује листа имена датотека. Да би изменио име датотеке корисник селекује текст на екрану а затим куца ново име датотеке.

Предности интерфејса за директну манипулацију су:

- корисници осећају контролу рачунара и нису њиме заплашени,
- време обучавања корисника је релативно кратко,
- корисници добијају непосредну повратну везу и грешке често могу да се открију и исправе врло брзо,

Један од најпростијих и најлакши за разумевање интерфејса са директном манипулацијом јесте интерфејс заснован на формуларима (обрасцима, енгл. forms). Кориснику се приказује формулар који треба да попуни. Поља у рачунару су означена именима која указују на информацију коју треба обезбедити. Слика 6.14 је пример интерфејса заснованог на формуларима који се може користити за прикупљање информација за библиотечки контакт са читаоцима.

Корисник уноси информацију која се захтева и прелази на следеће улазно поље користећи тастер < *Tab* > или бирајући поље помоћу миша. Са сваким пољем могу бити повезане контроле које проверавају да ли су унети подаци у коректном облику. Када има



Сл. 6.14: Интерфејс заснован на формуларима

релативно мало могућих улаза за неко поље, кориснику се може отворити мени са могућностима избора.

Интерфејси засновани на формуларима погодни су за примене где корисницима може да се да нека обука. Они реално нису погодни за повремене кориснике.

6.6.3 Модели интерфејса

Један начин да се оствари конзистентност корисничког интерфејса је да се дефинише конзистентни модел или метафора за интеракцију (међусобно деловање) корисник-рачунар. Модел корисничког интерфејса треба да буде аналоган неком моделу реалног света који корисник разуме.

Најпознатија метафора је метафора радне области (радне површине, радног простора, енгл. desktop) где кориснички екран представља радну област. Ентитети система представљени су на екрану помоћу икона.

Метафора радне области није погодна за подршку комплексних система интеракције које нуде апликације као што су едитори, системи база података итд. Ови системи одржавају метафору радне области до извесног степена, али је допуњују додавањем управљачког панела (дијалог прозора, енгл. control panel) који је графичка репрезентација система команди. Овај управљачки панел може бити проста пруга (линија) икона, где свака икона представља одређену команду. Притисак (енгл. clicking) на икону команде еквивалентан је избору команде из менија или уношењу (куцању) њеног имена на тастатури. Сложенији управљачки панели могу укључивати додатне објекте корисничког интерфејса као што су поља за приказ, клизачи, индикатори итд.

Управљачки панели могу садржати (укључивати) неколико врста ентитета за подршку интеракције са корисником:

1. Тастери (дугмад, енгл. buttons). - Бирање тастера (притисак на тастер) има за последицу иницирање (покретање једне акције);
2. Прекидачи (енгл. switches). - То може бити скуп позиција за конфигурисање система или превођење система из једног стања у друго;
3. Менији (енгл. menus). То је скуп тастера или прекидача који се могу учинити видљивим и могу се бирати (селектовати). Избор у менију неког назива изазива појављивање новог падајућег менија;
4. Индикатори (енгл. indicators). - Они се активирају да покажу да је нека акција у току;
5. Дисплеји или поља за приказ (енгл. displays). - То су области на панелу где се може приказати текстуална или графичка информација. Поља имају име и вредност;
6. Клизачи (енгл. sliders). - То су улазни уређаји који се користе за постављање улазне вредности из неког опсега. Слични су клизачима који контолишу хардвер, као што је контрола јачине звука на стерео систему. Корисник повлачи клизач дуж скале (опсега) да би поставио потребну вредност.