System Software

პროგრამული უზრუნველყოფა ეწოდება კომპიუტერული სისტემის ფუნქციონირებისათვის საჭირო კომპიუტერული პროგრამების, პროცესებისა და სათანადო დოკუმენტების ერთობლიობას.

კომპიუტერულ სისტემებში პროგრამული უზრუნველყოფა ძირითადად არის ინსტრუქციების ან ბრძანებების ნაკრები, რომელიც ეუბნება კომპიუტერს თუ რა უნდა გააკეთოს.

არსებობს ორი ტიპის პროგრამული უზრუნველყოფა: სისტემური და აპლიკაციური.

თანდართული პროგრამული უზრუნველყოფისგან განსხვავებით სისტემურ პროგრამებს არ შეუძლიათ პრაქტიკული ამოცანების გადაჭრა, ამის ნაცვლად ისინი უზრუნველყოფენ სხვა პროგრამების მუშაობას, მათთვის მომსახურების ფუნქციების და აპარატურული რესურსების მინიჭების საშუალებით.

სისტემური ხელსაწყოების არჩევანი დამოკიდებულია კომპიუტერის ოპერაციულ სისტემაზე, კომპიუტერის დანიშნულებაზე, მომხმარებელთა რაოდენობაზე, ორგანიზაციის მასშტაბზე, ქსელის არქიტექტურაზე და მრავალ სხვა თავისებურებებზე.

სისტემური პროგრამული უზრუნველყოფა შეიცავს ისეთ პროგრამებს რომელიც გამოიყენება კომპიუტერის სამართავად, როგორებიცაა ოპერაციული სისტემა, ფაილის მართვა და მყარი დისკის მართვის სისტემა. (DOS). System software წარმოადგენს პლატფორმას სხვა software-ებისთვის. მაგალითად შეგვიძლია მოვიყვანოთ: ოპერაციული სისტემა, ანტივირუს software, კომპიუტერის ენის მთარგმნელი და ა. შ. როგორც წესი მსგავსი software ები აღჭურვილია კომპიუტერის მწარმოებლის მიერ და დაწერილია დაბალი დონის ენაზე და გამოიყენება აპარატურულ აღჭურვილობასთან საკომუნიკაციოდ. მარტივარ რომ ვთქვათ system software ემსახურება ინტერფეისის სახით აპარატურულ აღჭურვილობასა და მომხმარებელს შორის კომუნიკაციას.

ყველაზე მნიშვნელოვან მახასიათებელს system software-ის წარმოადგენს:

1) სწრაფქმედება(უნდა იყოს მაქსიმალურად სწრაფი რათა უზრუნველყოს ეფექტური გარემო შესაბამისი software ისთვის.)

2) სისტემასთან სიახლოვე (უნდა იყოს უშუალო კავშირში აპარატურულ აღჭურვილობასთან.)

3)დაბალი დონის ენაზე დაწერილი(რათა პროცესორმა და სხვა აპარატურულმა აღჭურვილობამ შეძლოს წაკითხვა)

4)მრავალმხრივი (უნდა ქონდეს კავშირი როგორც აპარატურულ ისე პროგრამულ უზრუნველყოფასთან)

5)დაცული (რთული უნდა იყოს მასზე მანიპულირება, როგორც წესი მის შესაცვლელად საჭიროა პროგრამული ენა და მომხმარებლის ინტერფეისში შეუძლებელია.)

სისტემური პროგრამული უზრუნველყოფა მჭიდროდაა დაკავშირებული კომპიუტერთან და მის განუყოფელ ნაწილს წარმოადგენს. იგი შედგება: საბაზისო, სერვერული და ტესტური პროგრამული უზრუნველყოფისგან.

1)საბაზისო პროგრამული უზრუნველყოფა კომპიუტერში ახდენს ინფორმაციის დამუშავების პროცესის ორგანიზებას და გამოყენებითი პროგრამებისთვის ქმნის ნორმალურად ფუნქციონირებისათვის საჭირო გარემოს. მათ მიეკუთვნება ოპერაციული სისტემა.

2)სერვერული პროგრამული უზრუნველყოფა ეწოდება საბაზისო პროგრამული უზრუნველყოფის შესაძლებლობის გამაფართოვებელი პროგრამებისა და პროგრამული კომპლექსების ერთობლიობას. მასში გაერთიანებულია: ანტივირუსების პროგრამები(ფაილების, საქაღალდეების და დისკების არქივატორები), სარეზერვო კოპირების უტილიტები, ანტიფიშინგერები და ა.შ.

3)ტესტური პროგრამული უზრუნველყოფა განკუთვნილია კომპიუტერისა და კომპიუტერის ქსელის მუშაობის პროცესში წარმოშობილი შეცდომების დიაგნოსტირებისა და გასწორებისათვის. მას მიეკუთვნება დაპროგრამების ენები და სისტემები, აგრეთვე პროგრამების შექმნის პროცესის ავტომატიზებისთვის განკუთვნილი CASS საშუალებები.

**ოპერაციულის სიტემა**

System software ის ერთ-ერთ მაგალითს წარმოადგენს ოპერაციული სისტემა (OS) რომელიც მართავს კომპიუტერის პროგრამულ და აპარატურულ რესურსებს. როდესაც კომპიუტერი ირთვება პირველ რიგში ხდება ოპერაციული სისტემის ჩატვირთვა.

ოპერაციული სისტემა ასრულებს:

**1)მეხსიერების მართვა:** უზრუნველყოფს მეხსიერების მიწოდებას პროცესის მიერ მოთხოვნის შესაბამისად.

**2)პროცესორის მართვა:** გამოყოფს მეხსიერებას (RAM) პროცესისთვის, სანამ პროცესს სჭირდება.

**3)ფაილის მართვა:** განსაზღვრავს თუ რას რომელი რესურსი ჭირდება და ამის მიხედვით აწვდის.

**4)უსაფრთხოება:** ფაილებსა თუ პროგრამებზე წვდომის უფლებების კონტროლი.

**5)შეცდომების დაფიქსირება:** error მესიჯები.

**6)პროცესების განრიგი:** ოპერაციული სისტემა ადგენს განრიგს, კონკრეტული ალგორითმის მიხედვით, პროცესების სამართავად.

ოპერაციული სისტემა (Microsoft Windows, Mac OS, Linux), ისეთი დავალებების შესრულებით, როგორიცაა მეხსიერებასა და დისკებს შორის მონაცემების გადაცემა და დისპლეიზე გამომავალი მონაცემების გადმოცემა კომპიუტერის ნაწილებს ერთად მუშაობის საშუალებას აძლევს. ის უზრუნველყოფს პლატფორმას(აპარატურული უზრუნველყოფის აბსტრაქციის ფენას) მაღალი დონის პროგრამული უზრუნველყოფისა და გამოყენებითი პროგრამებისთვის.

ოპერაციული სისტემის მთავარი შემადგენელი ნაწილია ბირთვი. ის გამოყენებითი პროგრამებისთვის (მათ შორის ზოგიერთი სისტემური პროგრამული უზრუნველყოფისთვის) გამოყენებითი პროგრამირების ინტერფეისს და მოწყობილობის დრაივერების ინტერფეისს განსაზღვრავს.

მოწყობილობის დრაივერები, მათ შორის კომპიუტერის BIOS და ფიქსირებული პროგრამული უზრუნველყოფა კომპიუტერთან დაკავშირებული ან მასში ჩადგმული აპარატურის არსებითი ფუნქციების საშუალებას იძლევა.

სამომხმარებლო ინტერფეისი „მომხმარებლებს კომპიუტერთან ურთიერთქმედების საშუალებას აძლევს“. ეს ხდება ბრძანებების სტრიქონის ინტერფეისის (CLI) ან, 80-იანი წლების შემდეგ მომხმარებლის გრაფიკული ინტერფეისის გამოყენებით(GUI). ვინაიდან ეს ოპერაციული სისტემის ის ნაწილია, რომელთანაც მომხმარებელი პირდაპირ ურთიერთქმედებს, ის შეიძლება სისტემური პროგრამული უზრუნველყოფის ნაცვლად გამოყენებით პროგრამად ჩაითვალოს.

**Language Processor:** როგორც ვიცით system software-ს რომელიმე ენაზე დაწერილი კოდი გადაყავს მანქანურ კოდში და პირიქით. ეს კავშირი ხორციელდება language processor-ის მიერ. არსებობს 3 ტიპის language processor-ი: ასემბლერი, კომპილერი, ინტერპრეტატორი.

**კომპილერი:** ეს არის კომპიუტერული პროგრამა, რომელიც ერთი ენიდან მეორე ენაზე თარგმნის კოდს, ისე რომ კოდის შინაარსი არ იცვლება. კომპილერის ერთერთი მთავრი თვისებაა მოცემულ კოდში შეცდომების პოვნა და მათი გამოსწორება ან მითითება. არსებობს სხვადასხვა ტიპის კომპილერები მაგალითად:

cross-compiler (აწარმოებს სამუშაოს სხვა CPU-სთვის ან განსხვავებული ოპერაციული სისტემისთვის და არა იმაზე რაზეც თავად მუშაობს, მაგალითად კომპილერი შეიძლება მუშაობდეს კომპიუტერში მაგრამ ასრულებდეს კოდს რომელიც განკუთვნილია ანდროიდ სმარტფონისთვის).

Source-to-source compiler (თარგმნის მაღალი დონის ენიდან ისევ მაღალ დონის ენაზე, მაგალითად პითონიდან ჯავაზე. ასევე იყენებენ კოდის ძველი ვერსიის გასანახლებლად, მაგალითად თუ გადავაწოდებთ c++-ის რომელიმე ძველი ვერსიის კოდს, შესაძლებელია ამ კოდის უახლეს ვერსიაზე გადაყვანა.)

Decompiler (არის კომპილერის საწინააღმდეგო, ანუ დაბალი დონის კოდი გადაყავს მაღალი დონის კოდზე, დეკომპილერის მიერ შესრულებული ეს მოქმედება საკმაოდ კომპლექსურია და როგორც წესი არშეუძლია კოდის რეკონსტრუქცია ზუსტად გადმოიტანოს

მაგალითად ამ კოდის მანქანური ინსტრუქცია დეკომპილირებული IR ინსტრუქციაზე

mov eax, [ebx+0x04]

IR ზე გადავა როგორც

eax := m[ebx+4];

)

bootstrap compiler, compiler-compiler.

**ინტერპრეტატორი:** ეს არის კომპიუტერული პროგრამა, რომელიც პირდაპირ უშვებს კოდს, კომპილაციის გარეშე. ინტერპრეტატორი კომპილერისგან განსხვავებით კითხულობს ხაზებს სათითაოდ, გადაყავს მანქანურ კოდში და ასრულებს, პრობლემის აღმოჩენისთანავე წყდება პროცესი, თუმცა შეცდომის დაფიქსირებამდე არსებული შედეგები დაიბეჭდება. კომპილერს კოდი ჯერ გადაყავს მანქანურ ენაზე და შემდეგ უშვებს, ასე რომ გაშვების სიჩქარე უფრო სწრაფია.

**ასემბლერი:** ეს არის პროგრამა, რომელსაც ასემლერის ენაზე დაწერილი კოდი გადაყავს მანქანურ კოდში. ასემბლერი იღებს მარტივ ბრძანებებს, ოპერაციებს და გადაყავს ორობით კოდში პროცესორის ტიპის მიხედვით, შესაბამისად ასემბლერი არის მოწყობილობაზე დამოკიდებული ენა(machine-dependent) მაგალითად Intel 8085 და Intel 8086-ის არქიტექტურა განსხვავებულია.

**მოწყობილობის დრაივერი:** ეს არის ინსტრუქციების ნაკრები რომელიც უზრუნველყოფს კავშირს აპარატურასა და იმ პროგრამას ან ოპერაციულ სისტემას შორის რომელმაც უნდა გამოიყენოს ის. ყველა ფიზიკურ მოწყობილობას როგორებიცაა პრინტერი, მაუსი, როუტერი და ა.შ. სჭირდება დაივერი რათა დაუკავშირდეს კომპიუტერის სიტემას, ასე რომ ახალი მოწყობილობის დაერთების შემდეგ საჭიროა შესაბამისი დრაივერის ინსტალაცია რათა ოპერაციულმა სისტემამ იცოდეს თუ როგორ იმუშაოს მოცემულ მოწყობილობასთან.

მოწყობილობების დრაივერის გამოყენება სხვადასხვაგვარად ხდება ყველა ოპერაციულ სისტემაზე თუმცა ემსახურება ერთი და იმავე მიზანს .