

Odložišče (clipboard)

Odložišče oz. clipboard je eden od načinov prenosa podatkov (besedila, slik, itd.) znotraj in med uporabniškimi programi. Ponuja dve osnovni operaciji, tj. kopiranje (*copy*) in lepljenje (*paste*), ki ju sproži uporabnik preko bližnjic na tipkovnici (`Ctrl-C` in `Ctrl-V`) oz. drugih menijev znotraj posameznega programa. Poleg teh dveh se pogosto pojavi še operacija izrezovanja (*cut*), ki pa je pravzaprav samo kombinacija kopiranja in brisanja, zato je iz stališča sistemske programske opreme ni potrebno posebej obdelovati.

Kot uporabniki do odložišča najpogosteje dostopamo znotraj programov z grafičnim umesnikom (GUI), vseeno pa nekateri operacijski sistemi nudijo tudi orodja za dostop preko ukazne vrstice. Npr. na Windowsu je to ukaz `clip`, na Linuxu `xsel`, na MacOS pa `pbcopy` in `pbpaste`.

Poleg odložišča operacijski sistemi ponavadi ponujajo tudi druge načine prenosa podatkov med programi. Eden takih je povleci-in-spusti (*drag-and-drop*), ki je dokaj podoben odložišču, le da podatke "kopiramo" z vlečenjem (*drag*) in "prilepimo" s spustom (*drop*). Tudi na nivoju operacijskega sistema je *drag-and-drop* implementiran podobno kot odložišče, zato se mu v nadaljevanju ne bom posebej posvečal.

Osnovni principi delovanja

Odložišče si lahko predstavljamo kot medpomnilnik (buffer), dostopen vsem programom, v katerega lahko shranjujejo različne oblike podatkov in jih kasneje preberejo nazaj. Na njem je lahko shranjen največ en element naenkrat, ob vsaki operaciji kopiranja se predhodna vsebina odložišča pobriše. Poleg podatkov sistem beleži tudi trenutnega "lastnika odložišča", tj. okno oz. proces, ki je nanj zadnji shranil podatke.

Znotraj odložišča je lahko naenkrat shranjenih več oblik (formatov) podatkov. Pri tem ne gre za več ločenih kosov podatkov, ampak za zapis istih podatkov v večih alternativnih formatih. S tem poskrbimo, da je vsebina odložišča dostopna čim večjemu številu različnih programov.

Delo s odložiščem poteka v dveh fazah, tj. kopiranje in lepljenje. Spodaj sta predstavljena postopka v splošnem, v nadaljevanju pa bom predstavil še posebnosti pri posameznih operacijskih sistemih.

Kopiranje:

- Uporabnik znotraj programa sproži kopiranje, npr. preko bližnjice `Ctrl-C` na tipkovnici.
- Program sporoči sistemu, naj izprazni odložišče, in s tem postane lastnik odložišča.
- Na odložišče zapiše, katere oblike podatkov lahko ponudi (npr. besedilo, obogateno besedilo, sliko). V nekaterih primerih nanj zapiše tudi dejansko vsebino (npr. pri kopiranju manjših količin besedila), v drugih pa ne (če gre za t.i. *delayed rendering*).

Lepljenje:

- Uporabnik znotraj programa sproži lepljenje, npr. preko bližnjice `Ctrl-V`.
- Program iz odložišča prebere seznam oblik podatkov, ki so na voljo, in izmed njih izbere tisto, ki mu najbolj ustreza. Če ne podpira nobene od oblik, ki so na voljo, (npr. v urejevalnik navadnega besedila poskušamo prilepiti sliko), ne naredi nič.
- Od sistema zahteva vsebino željene oblike. Če lastnik odložišča za to obliko še ni zapisal dejanske vsebine (ker uporablja t.i. *delayed rendering*), mu sistem sporoči, da naj jo zapiše zdaj.
- Ciljni program iz odložišča prebere vsebino željene oblike. Pri tem se odložišče ne izprazni, lastnik odložišča pa ostane isti kot prej.

Razširitveni programi

Za večino operacijskih sistemov so na voljo t.i. *clipboard managers*, ki ponujajo različne razširitve sistemskega odložišča, kot so možnost shranjevanja večih elementov, hranjenje zgodovine, in podobno. Pri tem pa je potrebno omeniti, da ne gre za razširitve v pravem pomenu besede, vendar le za sledeči "trik":

- Ko uporabnik kopira podatke (npr. besedilo) v sistemsko odložišče, to *clipboard manager* zazna, in shrani vsebino sistemskega odložišča v svojo lokalno shrambo.
- Ko uporabnik v takem *clipboard managerju* izbere element, ki ga želi prilepiti, ga program zapiše nazaj v sistemsko odložišče.

Iz stališča sistema in ostalih programov se odložišče obnaša enako kot prej, teh "več shranjenih elementov" oz. zgodovina pa drugim kot *clipboard managerju* ni vidna.

Primerjava implementacij

Ker se odložišče v večini primerov uporablja v programih z grafičnim vmesnikom, je iz praktičnih razlogov ponavadi implementirano kot del okenskega sistema (*windowing system*). V nadaljevanju bom predstavil dve taki implementaciji, in sicer v operacijskem sistemu Windows ter v okenskem sistemu X11 (ki ga uporablja Linux in ostali Unixu podobni operacijski sistemi).

Windows

Odložišče je na sistemu Windows implementirano znotraj modula `win32s.sys` (skupaj s preostankom okenskega sistema), programi pa do njega dotopajo preko knjižnice `user32.dll`.

Kopiranje:

- Program s klicem `OpenClipboard` odpre odložišče in ga s klicem `EmptyClipboard` pošisti, s čimer postane njegov lastnik. Ob tem sistem predhodnemu lastniku pošlje zahtevo `WM_DESTROYCLIPBOARD`.
- Z enim ali več klicev `SetClipboardData` zapiše obliko in vsebino podatkov v vseh oblikah, ki jih podpira. Pri tem lahko vsebino nastavi na `NULL` in poda samo njeno obliko, s čimer sistemu pove, da želi uporabiti t.i. *delayed rendering*.
- S klicem `CloseClipboard` zapre odložišče, s čimer zaključi pisanje vsebine in omogoči dostop drugim programom.

Lepljenje:

- Program s klicem `OpenClipboard` odpre odložišče.
- S klicem `EnumClipboardFormats` prebere seznam oblik, ki so na voljo.
- S klicem `GetClipboardData` prebere vsebino željene oblike. Če je bil pri tej obliki uporabljen t.i. *delayed rendering*, sistem pošlje lastniku zahtevo `WM_RENDERFORMAT`. Lastnik se nanjo odzove tako, da s klicem `SetClipboardData` zapiše vsebino željene oblike.
- Po koncu branja s klicem `CloseClipboard` zapre odložišče.

Oblike vsebine so zapisane kot cela števila in se delijo v tri skupine:

- Standardne oblike: definirane kot številske konstante (npr. `CF_UNICODETEXT` za navadno besedilo, `CF_DIB` za slike).

- Registrirane oblike: definirane kot nizi poljubne oblike (v praksi se uporabljajo MIME tipi, končnice datotek, ali pa kaj drugega). Pred uporabo jih je potrebno registrirati s klicem `RegisterClipboardFormat`, da jim sistem priredi enolično oznako.
- Zasebne oblike: namenjene so kopiranju podatkov znotraj enega programa. Številске oznake zanje si program prosto izbere iz intervala med `CF_PRIVATEFIRST` in `CF_PRIVATELAST`.

X11 (Linux)

Odložišče je v okenskem sistemu X11 implementirano preko koncepta "izborov" (*selections*). V standardu so definirani trije izbori: `PRIMARY`, `SECONDARY`, `CLIPBOARD`, programi pa lahko definirajo tudi izbore z drugimi imeni (kar se uporablja za npr. *drag-and-drop*). Vsak izbor si lahko predstavljamo kot eno odložišče, le da sistem zanj hrani samo lastnika (ne pa tudi dejanske vsebine ali oblike podatkov).

Standard sam sicer ne pove, kako naj bi programi uporabljali posamezne izbore, v praksi pa je preko izbora `CLIPBOARD` implementirano to, kar uporabniki štejejo za "klasično" odložišče.

Kopiranje:

- Uporabnik znotraj programa sproži kopiranje.
- S klicem `XSetSelectionOwner` se program javi sistemu kot lastnik izbora. Pri tem prejšnji lastnik izgubi lastništvo, sistem pa mu pošlje zahtevo `SelectionClear`.
- Drugih podatkov (kot so vsebina in oblika) še ne pošlje sistemu.

Lepljenje:

- Uporabnik znotraj programa sproži lepljenje.
- S klicem `XGetSelectionOwner` program poišče trenutnega lastnika izbora.
- S klicem `XConvertSelection` od lastnika izbora zahteva vsebino izbora. Pri tem mora podati željeno obliko vsebine (npr. `UTF8_STRING` za navadno besedilo) in ciljno okno (nanj bo lastnik izbora zapisal podatke).
- Sistem pošlje lastniku izbora zahtevo `SelectionRequest`. Ko jo ta prejme, na ciljno okno s klicem `XChangeProperty` zapiše vsebino odložišča željene oblike ter mu pošlje zahtevo `SelectionNotify`.
- Ciljni program prejme zahtevo `SelectionNotify` in s klicem `XGetWindowProperty` prebere vsebino.

Z izjemo nekaj standardnih imen oblik (kot je npr. `UTF8_STRING` za navadno besedilo) se po dogovoru za poimenovanje oblik vsebine uporabljajo MIME tipi (npr. `text/html` za obogateno besedilo v formatu HTML).

Poleg zgornjih oblik se v praksi uporablja še oblika `TARGETS`. Preko nje ciljni program od lastnika odložišča zahteva seznam oblik vsebine, ki so trenutno na voljo, na podlagi česar se odloči, vsebino kakšne oblike bo zahteval.

Poleg klasičnega odložišča pa večina programov podpira tudi izbor `PRIMARY`. Vanj programi "zapišejo" trenutno označeno besedilo, ne da bi ga uporabnik posebej kopiral tja, kasneje pa ga lahko uporabnik prilepi podobno kot vsebino klasičnega odložišča. Po dogovoru lahko lepljenje iz tega izbora uporabnik sproži s srednjim klikom na miški.

Delo z odložiščem v jeziku Java

Razredi za delo z odložiščem so v Javi del paketa `java.awt.datatransfer`, branje in pisanje v odložišče pa poteka preko razreda `Clipboard` v tem paketu.

Do vsebine odložišča dostopamo preko klicev `getAvailableDataFlavors` (vrne seznam oblik, ki so na voljo) in `getData` (vrne podatke željene oblike).

Novo vsebino zapišemo na odložišče s klicem `setContents`, ki mu podamo vsebino (kot objekt tipa `Transferable`) in lastnika. Lastnik je v tem primeru objekt, ki implementira vmesnik `ClipboardOwner`, preko njega pa lahko program izve, kdaj je izgubil lastništvo nad odložiščem.

Za oblike podatkov je uporabljen izraz "okus" (*flavor*). Predstavljeni so z razredom `DataFlavor`, sestavljeni pa so iz MIME tipa in imena javanskega razreda. MIME tip pove obliko podatkov, ime razreda pa pove tip objekta, ki ga vrne `getData` (to je lahko `String`, `byte[]`, ali pa kaj drugega).

Viri in literatura

[https://en.wikipedia.org/wiki/Clipboard_\(computing\)](https://en.wikipedia.org/wiki/Clipboard_(computing))

<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/clip>

<https://manpages.ubuntu.com/manpages/bionic/man1/xsel.1x.html>

<https://ss64.com/osx/pbcopy.html>

<https://docs.microsoft.com/sl-si/windows/win32/dataxchg/clipboard>

<https://freedesktop.org/wiki/Specifications/ClipboardsWiki/>

<https://www.jwz.org/doc/x-cut-and-paste.html>

<https://www.uninformativ.de/blog/postings/2017-04-02/0/POSTING-en.html>

<https://www.codeproject.com/Articles/43694/Forbidding-the-Clipboard-for-the-Specified-Process>

<https://www.codeproject.com/Reference/1091137/Windows-Clipboard-Formats>

<https://docs.oracle.com/javase/7/docs/api/java/awt/datatransfer/package-summary.html>