

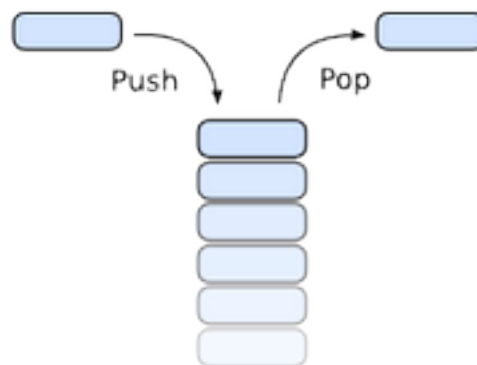
Zásobník (LIFO)

Je dynamická datová struktura s těmito dovolenými operacemi:

CREATE(S) - vytvoří prázdný zásobník S,
TOP(S) - vrátí prvek na vrchu zásobníku S,
POP(S) - vyhodí prvek na vrchu zásobníku S,
PUSH(x, S) - vloží prvek x na vrch zásobníku S,
EMPTY(S) - vrátí TRUE, je-li zásobník prázdný, jinak vrátí FALSE.

Zásobník (anglicky STACK) se také označuje jako struktura LIFO (Last In, First Out – poslední dovnitř, první ven). Je to posloupnost prvků nějakého typu, přičemž přístup k této posloupnosti je pouze na jednom jejím konci, který se nazývá vrchol (anglicky TOP) zásobníku. Obsah zásobníku lze měnit jen na tomto jednom konci pomocí dvou základních operací:

- Přidání prvku na vrchol zásobníku (PUSH)
- Odebrání prvku z vrcholu zásobníku (POP)

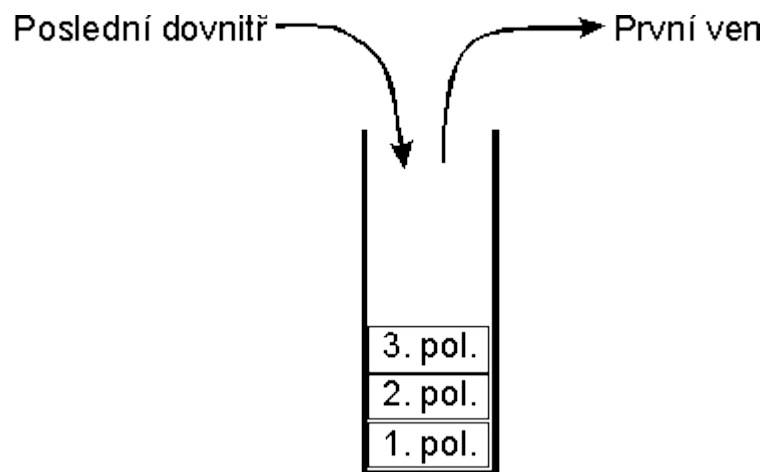


Opakem zásobníku LIFO je fronta FIFO (First In, First Out – první dovnitř, první ven)

Zásobník lze přirovnat k zásobníku nábojů v pistoli nebo samopalu. Náboje jsou přesouvány do nábojové komory v opačném pořadí, než byly do zásobníku vloženy. V jednom okamžiku máme k dispozici pouze horní náboj, nebo je zásobník prázdný. Ke spodním nábojům se lze dostat jen vyjmutím předchozích nábojů.

Ukazatel na aktuální prvek v zásobníku (posledně vložený) se nazývá vrchol zásobníku (anglicky STACK POINTER). Opakem je dno zásobníku. Operace vložení do zásobníku se tradičně nazývá PUSH a vyjmutí se nazývá POP. Jako třetí se u zásobníku implementuje dotaz EMPTY, který indikuje prázdnost zásobníku. Navíc se někdy přidává dotaz TOP, který vrací prvek na vrcholu zásobníku, aniž by ho vyjmul (nedestruktivní varianta POP). Pokud provedeme operaci POP na prázdném zásobníku, nastává chyba tzv. „podtečení“ (anglicky UNDERFLOW). Zásobník má teoreticky neomezenou kapacitu. Pokud ji omezíme např.

velikostí přidělené paměti, a nelze již přidat další prvek, nastává opět chyba tzv. „přetečení“ (anglicky OVERFLOW).



Příklady, kde všude v běžném životě se lze setkat se zásobníkem LIFO:

U nás doma:

Už před rokem jsme se přestěhovali do většího bytu, a přesto pořád máme spoustu věcí v krabicích. A v nich hlavně noty, těch máme spoustu. Takže systém LIFO zde funguje dokonale. Když potřebuje žena najít nějaké noty, musí nejprve spoustu vyndat, až se dostane k těm požadovaným. Uvažoval jsem, že bychom mohli odstříhnout dno a udělat jedním šmahem ze zásobníku LIFO frontu FIFO, ale nikdo se mnou nesouhlasil. Tak už mi zbývá jen to Delphi, tam to mohu zkusit jak je libo.

V restauraci:

V restauračních zařízeních můžeme na systém LIFO narazit kdykoliv. Stačí, když nám číšník začne nosit jídlo v opačném pořadí. Znáte to sami. Nejdříve vám přinesou kompot, pak hlavní jídlo a teprve potom polévku. No a vy to samozřejmě sníte v opačném pořadí, tak jak se to běžně dělá, tedy nejprve polévka, pak hlavní jídlo a nakonec kompot. No není to princip zásobníku?

Zaživací trakt:

...pokračování z restaurace. Zde to nebudu moc rozvádět, však to znáte sami. Pokud to jde nazpět vrchem, jedná se o LIFO, no a pokud to jde spodem, tak už je to FIFO.

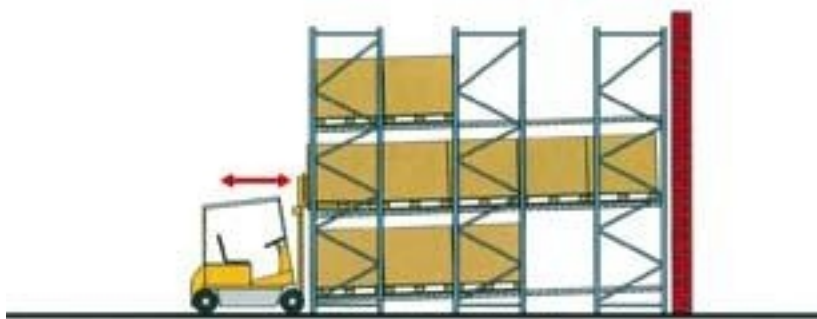
Funkce VoIP telefonní ústředny:

Stahování hovoru z jiných telefonů - uživatel si může převzít hovor z jiného pobočkového VoIP telefonu. V případě více příchozích hovorů platí pravidlo LIFO - poslední příchozí hovor bude převzat jako první a první příchozí hovor jako poslední.

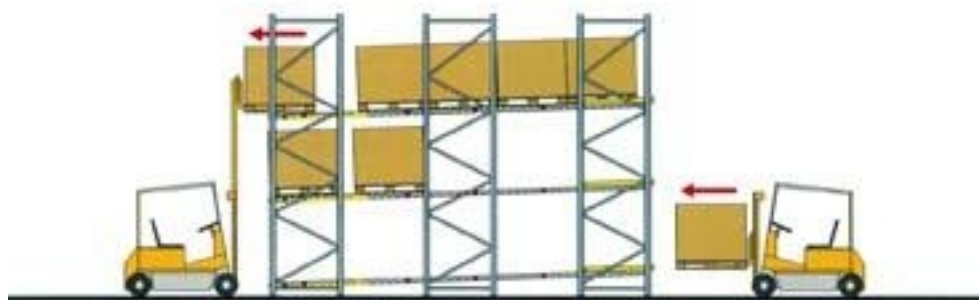
Regály ve skladu:

Tohle jsem se dočetl na netu, tak dělám chytrého. Existují dva druhy regálů:

1. **Push-back regály** – Mají pouze jednu obslužnou rovinu, zakládání i odebrání palet je pouze z jedné strany. Při plnění kanálů paletami zatlačuje nebo brzdí manipulační vozík i palety, které jsou již založené. Po odebrání poslední palety sjíždí ostatní do místa odběru. Skladování palet v push-back regálech je dle principu LIFO.



2. **Spádové regály** - Jejich použití je všude tam, kde je vyžadováno skladování dle principu FIFO. Mají dvě obslužné roviny, z jedné strany se palety zakládají do kanálů, na straně druhé se pak z kanálu odebírají. Konstrukce dráhy je navíc opatřena koncovým oddělováním ostatních palet v kanálu od poslední, pro její bezpečné odebrání.



Ve skladovém účetnictví:

„Last In., First Out“ = „poslední do skladu., první ze skladu“ Poslední cena použitá při ocenění přírůstku zásob se použije jako první cena při ocenění úbytku zásob. Při inflaci, tj. vyššímu růstu cen vede k vykazování nižší hrubé marže tedy nižšímu výsledku hospodaření ve výsledovce na úkor přesnějšího ocenění zůstatku zásob v rozvaze (stav je podhodnocen)

Zmrzlina

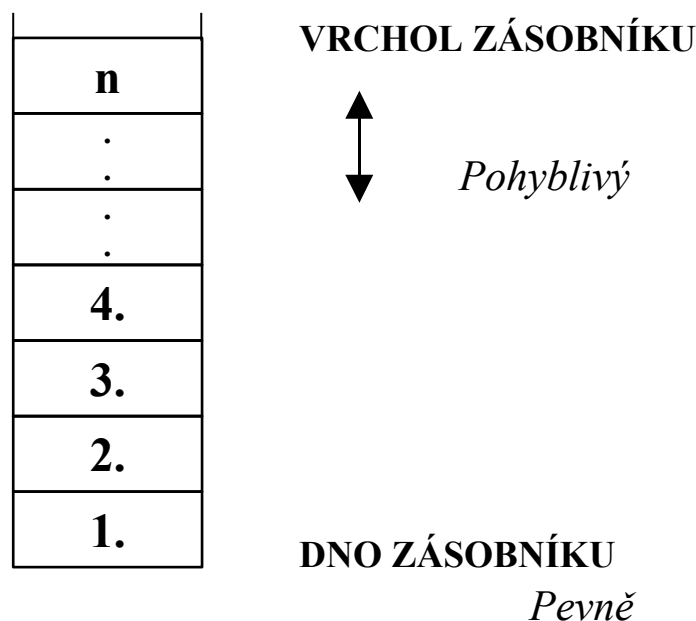
(hackerská legenda, převzatá z internetu)

Za starých časů jste si mohli dát v cukrárně zmrzlinu – třeba jahodovou, oříškovou, pistáciovou a vanilkovou, dostali jste do kornoutu čtyři kopečky zmrzliny a mohli jste postupně lízat jahodovou, oříškovou, pistáciovou a vanilkovou zmrzlinu.

Jenže pak si nějací chytráci uvědomili, že kornout není hardwarově fronta, ale zásobník, takže fronta je patrně emulována na úrovni prodavačky, která tím pádem musí mít vyrovnávací paměť, kde si uloží všechny objednané zmrzliny, aby je mohla naskládat do kornoutu v opačném pořadí. Rychle zjistili, že prodavačky nejsou chráněny proti buffer overflow, a brzy vymysleli první exploity. Stačilo jim pak přijít do cukrárny, objednat si například 65 libovolných zmrzlin a pak ještě oříškovou, citronovou, dvě kávové a znova oříškovou a prodavačka vyplivla rootshell a mohli se nacpat zmrzliny, kolik chtěli, a někteří pak ještě sežrali nějaké ty indiánky, rolády, trubičky či kreky (podle nich se také těmto chytrákům začalo říkat krekři, anglicky crackers).

Bylo jasné, že zvětšením paměti prodavaček se nic nevyřeší (navíc byly tehdy paměti do prodavaček neuvěřitelně drahé) a i zákaz nošení jmenovek se ukázal jako nepříliš účinné řešení – krekři poznali prodavačku podle obličeje, a když ne, tak prostě vyzkoušeli víc kombinací zmrzlin. Řada cukrářů tehdy investovala do inteligentních prodavaček, které jakékoli objednávky nad deset kopečků ignorovaly a hlásily je šéfovi, ale řada zákazníků si stěžovala, že si už nemůžou normálně vzít patnáct zmrzlin do termosky a odnést si je domů, ale musí to pořád řešit s šéfem nebo si je objednávat po částech.

Nakonec to cukráři vzdali a sebrali prodavačkám veškerou paměť a změnili algoritmus ukládání zmrzliny do kornoutu tak, že se začala ukládat ve stejném pořadí, jak přicházejí požadavky. Od té doby jsou zmrzliny LIFO a když si objednáte jogurtovou a karamelovou, dostanete místo toho karamelovou a jogurtovou. Taky můžete jak dlouho chcete otravovat prodavačky, jestli budou zase někdy mít kávovou zmrzlinu, ale ony to vždycky okamžitě zapomenou.



Implementace zásobníku pomocí lineárního seznamu:

```
Type Pzas = ^zas;  
  zas = record cislo: Integer;  
    dalsi: Pzas  
  end;  
  
var vrch: Pzas;  
  
function EMPTY: Boolean;           //EMPTY - vrátí TRUE, je-li zásobník prázdný, jinak vrátí FALSE.  
begin  
  result:=vrch=nil  
end;  
  
function TOP: Pzas;               //TOP - vrátí prvek na vrchu zásobníku Pzas  
begin  
  result:=vrch  
end;  
  
procedure TForm1.Button1Click(Sender: TObject);           //prázdný zásobník  
begin  
  vrch:=nil;  
end;  
  
procedure TForm1.Button2Click(Sender: TObject);           //PUSH - vloží prvek q na vrch zásobníku Pzas  
var q: Pzas;  
begin  
  new(q);  
  if EMPTY then  
    q^.cislo:=1  
  else  
    q^.cislo:=vrch^.cislo+1;  
    q^.dalsi:=vrch;  
  vrch:=q;  
end;  
  
procedure TForm1.Button4Click(Sender: TObject);           //POP - vyhodí prvek q z vrchu zásobníku Pzas  
var q: Pzas;  
begin  
  if not EMPTY then  
    begin  
      q:=vrch;  
      vrch:=vrch^.dalsi;  
      dispose(q)  
    end;  
end;
```