prikazy skoku: break - vyskoci z jednoho cyklu continue - vrati se na podminku goto - return - vyhoda oproti break ze pouzit vice zanorenych cyklech a vyskocim ze vsech datovy typ ukazatel, alokace pameti alokace pameti: virtuarni pametovy prostor a ktera je 2 na 64 do UPP se zapise code a stat ktery mame pouze na cteni a potom nasleduje heap a stack(zasobnik) do kterych se uklada do stacku(ktery je omezen velikosti nebo do heapu int example = 51; zapisuje se do stacku

novy datovy typ int * ptr; -ukazatel muze ukazovat kamkoliv a musime si dat pozor aby neukazoval na jinou hodnotu a nebo nekam kde nemuzeme zapisovat ci cist -¿ ukazatel int * ptr nesmi byt nulovy -¿ segmentatovy fault int *ptr = example -¿ do ukazatele ptr nam uloz adresu hodnoty example int - nam urcuje co ukazatele bere jako pamet co patri te hodnoty *ptr -¿ 51 ptr -¿ adresa v pameti bazovy$_t yp * jmenopromenebazovy_t yp - urcujevelikostodkazovanepametirefencnivraciadresupromeneint * ptr = example * derefencivratihodnotunapameti- > inty = *px * ptr =$ 123$ziskejhodnotupametinaktereseukazujeukazatelaprepisthodnotyazapistonamistonaktereukazovalukazat$
$qNULLukazatelpouzetoprotestovatinepouzivatatestovatjestliukazatelNUllukazatelbytmelbytjasnouabezpe$
$nelzemenitsamotnouadresualetapametnakteroujeodkazovanojoint *$
$constCP = iukazatelnakonstantu-lzezmenitadresualepametkterajenaadresenejdemenitconstint*$
pci

Obecny ukazatel Typovy ukazatel umoznuje typovou kontrolu ale obecny ne daji nam kus pameti a hrej si

Ukazatel na ukazatel - na tvorba slozitejsich dynamickych datovych struktur - ulozime si adrseu na ktere lezi ukazatel ktery ukazuje na jinou hodnotu pokud chceme hodnotu int ** ppi = pi Dynamicka alokace pameti zasobnik + hromada na hromade je nutno pouzit ukazatl jelikoz nejde primo vytvorit pojmenovanou promennou a programator je zodpovedny za spravnou alokaci i dealokaci pamet funkce malloc z stdlib.h void *malloc(size$_t size)castosektomupouzivasizeoffalkujepametnahromadeavratiukazatelsizepocatalokovanychb$
$pi = malloc(sizeoff(int))) == NULL)returnERR_MALLOC;voleudelejsinatocvicenisizeoffjeoperatorresi$
$jehoztrata- > memoryleak- > ztracenoupametnelzeziskatzpetuvolnenipametmusimepametuvolnit-$
$free()int*pi = malloc(sizeof(int))free(pi)jednomalloc()jednofree()Datovytyppolepolejestejnehotypupoton$
$zodpovednyjeprogramatorbuffer-overrflowjakobezpecnostichybakdyzjemimopoleamuzemesiprepsatdata$