

```

struct Uzel { int Typ;
              union { struct { Uzel *prvni,*druhy,*treti,*ctvrty; } z;
              int Cislo;
              const char *Retez;
              int *Adresa; } z; };

Uzel *Koren=0; // Kořen stromu interní formy

int Interpr(const Uzel *u) // Interpretace interní formy
{
    if (u==NULL) return 0;

#define prvni    u->z.z.prvni
#define druhy    u->z.z.druhy
#define tretí    u->z.z.treti
#define ctvrty   u->z.z.ctvrty

    switch (u->Typ) {
        case 0: Interpr(prvni); Interpr(druhy);
                return 0;

        case '=': return *prvni->z.Adresa = Interpr(druhy);

        case P_DELEN: { int d=Interpr(druhy);
                        if (d==0) { printf("\nDeleni nulou\n"); abort(); }
                        return *prvni->z.Adresa/=d; }

        case PRINT: if (prvni->Typ!=RETEZ)
                        printf("%i",Interpr(prvni));
                    else if (druhy)
                        printf(prvni->z.Retez,Interpr(druhy));
                    else printf(prvni->z.Retez);
                    return 0;

        case FOR: Interpr(prvni);
                    while (Interpr(druhy)) { Interpr(ctvrty);
                                                Interpr(treti); }
                    return 0;

        case IF: if (Interpr(prvni)) Interpr(druhy);
                 else Interpr(treti);
                 return 0;

        case INKREM: if (prvni) return ++*prvni->z.Adresa;
                    return (*druhy->z.Adresa)++;

        case '!!': return !Interpr(prvni);

        case '<': return Interpr(prvni) < Interpr(druhy);

        case OR: return Interpr(prvni) || Interpr(druhy);

        case '-': if (druhy) return Interpr(prvni) - Interpr(druhy);
                 return -Interpr(prvni);

        case '/': { int d=Interpr(druhy);

```

```
        if (d==0) { printf("\nDeleni nulou\n"); abort(); }
        return Interpr(prvni)/d; }
case CISLO: return u->z.Cislo;
case PROMENNA: return *u->z.Adresa;
default: printf("\nNeznama operace: %i\n",u->Typ); abort();
    }
}
```

```
extern FILE *yyin;

int yyparse();

char *Source[81]; // cesta k souboru se zdrojovým programem µC
                  // je zadána na příkazovém řádku při volání interpretu

if (!(yyin=fopen(Source,"rt")))
    { /* neotevřel se zdrojový soubor */ }

Flush();
LexInic();

int parse=yyparse();

fclose(yyin);

if (Chyby || parse) exit(0);

Interpr(Koren);
```