

JEDNODUCHÝ DOKUMENTAČNÍ SYSTÉM

Ing. Bohumil Krejčířík

Při tvorbě a realizaci rozsáhlejších projektů je vhodné mít k dispozici prostředek, který zachycuje to, co bylo doposud vytvořeno, umí o projektu podat potřebné informace v pořádkovaném členění a v konečné fázi poskytnout nezbytnou dokumentaci. Rovněž je žádoucí, aby takový prostředek byl z hlediska využívání pokud možno co nejjednodušší a nezatěžoval uživatele příliš velkými nároky na svou obsluhu. Dále popisovaný návrh dokumentačního systému se snaží vyhovět uvedeným požadavkům. Poměrně jednoduchými požadavky umožňuje dosáhnout žádaných efektů.

1. Technické zabezpečení

Předpokládá se pořizování a vedení veškeré dokumentace na počítači. Jednotlivé prvky dokumentačního systému jsou uloženy jako členy knihovny na disku, které bude dále nazývána knihovnou zdrojových textů. Členy jsou uloženy formou znakových vět pevné délky 80 bytů, délka bloku je 800 bytů. Manipuluje se s nimi vhodným textovým editorem, například na počítači EC1032 pod operačním systémem OS/JS lze použít TSO nebo CBJE. Mimo knihovny zdrojových textů se vyřadují dva trvalé soubory s příjímým přístupem. Jeden slouží pro definici struktury dokumentačního systému a na disku 30 Mb spotřebuje maximálně 30 stop. Další slouží pro realizaci referencí, jeho velikost je závislá na předpokládaném rozsahu dokumentace, avšak i u rozsáhlejších projektů spotřebuje na stejném disku maximálně 10 až 20 stop. Všechny vstupy lze směřovat na řádkovou tiskárnu. Interaktivní přístup k tomuto dokumentačnímu systému je však vhodnější.

2. Struktura dokumentačního systému

Dokumentační systém se skládá z prvků - předmětů popisu /program, obrazovka, datová struktura apod./. Každý prvek je popsán jedním nebo více odstavci, které mohou být pro daný prvek povinné nebo volitelné. Tyto odstavce jsou saváděny jako členy knihovny zdrojových textů pod jménem a následující strukturou:

ABCXXXI, kde

A je symbol subsystému v rámci projektu

B je symbol skupiny úloh v rámci subsystému **A**

C je symbol prvku dokumentačního systému

XXI je pořadové číslo výskytu prvku **C** /nebo jinými znaky provedené rozlišení výskytu prvku **C**, např. programu/

I je symbol odstavce, který popisuje prvek **C**

Každému prvku je v rámci dokumentačního systému přiřazen jednoznačný identifikátor **ABCXXXI** a je popsán předepsanými odstavci, kterým je rovněž přiřazen jednoznačný identifikátor **ABCXXXI**, pod kterým je uložen v knihovně zdrojových textů.

Mezi prvky dokumentačního systému existují logické vazby, jde o hierarchické uspořádání, kdy prvek vyšší úrovně obsahuje odkazy na prvky nižší /nebo stejné/ úrovně. Například popis úlohy obsahuje odkazy na popisy použitých programů, popis programu se odkazuje na popisy použitých souborů, ty zase na popisy datových struktur atd. Vazby mezi jednotlivými prvky mají tedy obecně strukturu několikanásobného stromu. Tyto vazby jsou obsaženy v odstavci se symbolem **A**, který je přiřazen každému prvku a zavedení tohoto odstavce je prvek definován. Texty se v tomto odstavci zapisují vždy od prvního sloupce, je-li v prvním sloupci znak * /hvězdička/ je celý řádek chápán jako poznámka. Ostatní řádky obsahují od prvního sloupce identifikátory prvků, na které existuje z popisovaného prvku odkaz. Text od sedmého znaku je chápán jako poznámka. Pro větší přehlednost je vhodné jednotlivé dráhy prvků odlišit poznámkami, např.:

*PROGRAM PRO VÝPOČET STAVU ZÁSOB

#10. 2. 1988

*soubory

EBAC01

EBAC02

EBAC03 pomocný soubor

*datové struktury

EBFC01

EBFC05

*obrazovky

EBWC75

Nyní následuje návrh symboliky dokumentačního systému. Tato symbolika je použita jak v předchozím případě, tak i nadále. V záhlaví je vždy uveden symbol prvku, dále pak symboly odatavců, které ho popisují. U odatavců je uvedeno, zda je povinný.

A = soubor na disku

A - odkazy na formáty datových vět. Povinný.

B - popis účelu souboru. Povinný.

C - popis organizace souboru, předpokládaný rozsah. Povinný.

D - popis způsobu aktualizace souboru. Povinný.

G = soubor, který je součástí báze dat

má stejné odatavce jako prvek A

M = soubor na magnetické páse

má stejnou strukturu jako prvek A

F = datová struktura /věta/

A - pouze první dva řádky /pojmenování a datum vytvoření/. Povinný.

B - popis účelu datové struktury. Povinný.

C - datová struktura v příslušném programovacím jazyce. Slouží pro kopírování do programů. Povinný.

D - popis významných položek, odkazy na číselníky. Nepovinný.

W = obrazovka

A - pouze první dva řádky. Povinný.

B - popis účelu obrazovky. Povinný.

C - grafická podoba obrazovky, rekapitulace délek polí. Povinný.

D - znaková struktura obsahující předášená pole obrazovky. Slouží pro kopírování do programů. Povinný.

J = job

A - odkazy na použité programy. Povinný.

B - popis účelu jobu. Povinný.

C - struktura jobu. Popis jednotlivých kroků. Povinný.

D - provozní údaje /periodicita, region, návratové kódy, havarijní postupy/. Povinný.

mezera - zdrojový text jobu. Povinný.

S = sestavá

A - pouze první dva řádky. Povinný.

B - popis účelu a použití sestavy. Povinný.

C - provozní údaje /periodicita, předpokládaný dosah, druh papíru, distribuce/. Povinný.

H = zpráva_operátorovi

A - pouze první dva řádky. Povinný.

B - popis stavby zprávy. Povinný.

C - reakce operátora /odpovědi/. Povinný.

P = procedura_JGL

A - odkazy na programy. Povinný.

B - popis účelu procedury. Povinný.

mezera - text JGL příkazu. Povinný.

X = pracovní_postup

A - odkazy na programy, joby, procedury. Povinný.

B - popis účelu pracovního postupu. Povinný.

C - popis pracovního postupu. Povinný.

Z = program

A - odkazy na soubory, obrazovky, podprogramy, sestavy. Povinný.

B - popis účelu programu. Povinný.

C - algoritmus. Použije se jako programové zadání. Povinný.

D - prostředí ke spuštění. Povinný.

F - popis parametrizace. Nepovinný.

G - návod k práci s programem. Lze využít pro "HELP funkci" při interaktivním přístupu. Nepovinný.

mezera - zdrojový text programu. Povinný.

R = podprogram

má stejné odstavce jako program /Z/

U = uživatel

A - odkazy na pracovní postupy. Povinný.

B - popis začlenění uživatele do subsystému. Povinný.

C - popis činnosti uživatele, harmonogram práce. Povinný.

D - menu pro interaktivní zpracování. Nepovinný.

C = volný text

A - pouze první dva řádky. Povinné.

Tento odstavec obsahuje pouze volný text, který se využije při tisku dokumentace.

Všechny prvky mají nepovinný odstavec K - poznámka.

Uvedený výčet prvků a odstavců se může pro některé projekty ukázat jako nevyhovující. Především lze začlenit i popisy neautomatizovaných činností uživatelů, to zřejmě záleží na povaze projektu. K tomu ještě je třeba přihlížet při definici struktury dokumentačního systému pro konkrétní projekt.

Dokumentační systém by měl mít vestavěný následující mechanismy:

- definice struktury dokumentačního systému
- test kompatibility dokumentačního systému
- vytvoření indexu knihovny zdrojových textů
- tvorba seznamů, výskytů a odkazů prvků
- řízení tisk dokumentace

3. Definice struktury dokumentačního systému

Struktura dokumentačního systému je uložena v souboru a přístupným způsobem, který se skládá z vět následující struktury:

1 definice

2 symbol prvku /jeden znak/

2 název prvku /55 znaků/

2 tabulka odstavců /58 prvků tabulky/

3 indikace povinnosti odstavce /logická proměnná/

3 symbol odstavce /jeden znak/

3 název odstavce /55 znaků/

Pokud použijeme jako symboliku znaky abecedy, čísla a mezeru pak vystačí zhruba 58 těchto vět. Klíčem k přístupu k větě v tomto souboru je symbol prvku a symbol odstavce. Založení takového souboru je definována struktura dokumentačního systému. Jeho aktualizací, nejlépe interaktivní lze pak tuto strukturu obměňovat. Další jednoduchý program slouží pro výpis struktury dokumentačního systému.

4. Test kompletnosti dokumentačního systému

Účelem testu je odhalit nerovnosti mezi definicí struktury dokumentačního systému a realitou v knihovně zdrojových textů a tím přispět k postupnému kompletování dokumentace. Porovnává se abecedně seřazený adresář knihovny zdrojových textů, vlastní knihovna a soubor, kde je definována struktura dokumentačního systému. S ní se porovnávají odkazy v odstavcích A. Tím lze získat následující přehled:

- a/ prvky zdrojové knihovny, které nejsou definovány
- b/ prvky, na něž je odkaz v odstavcích A a nejsou ve zdrojové knihovně zavedeny
- c/ odstavce, které jsou definované jako povinné a nejsou ve zdrojové knihovně
- d/ prvky a odstavce, které jsou ve zdrojové knihovně a nejsou definovány.

Uvedené programy, především práce s knihovnou je nejlépe realizovat v Pl/1 nebo Assembleru.

5. Vytvoření indexu knihovny zdrojových textů

Pro výběry a tisk přehledů je nutno vytvořit tabulku /index/ vzájemných odkazů jednotlivých prvků dokumentačního systému. Tato tabulka je vytvářena na základě odstavců A tím způsobem, že se konstruuje dvojice:

Identifikace prvků, Identifikace odkazů a inverzní dvojice

Identifikace odkazů, Identifikace prvků.

Každé této dvojici je pak nutno ještě přiřadit jeden znak, aby se rozlišil směr přímý /P/ od inverzního /I/.

Věta má tedy následující strukturu: 1 index,

2 identifikátor prvku

2 směr

2 identifikátor odkazu

Identifikace jednotlivých prvků se do uvedených proměnných sepíše tak, že třetí znak /symbol prvku/ se zasadí za první znak. Je-li například v odstavci A prvku EBZ001 odkaz na prvky EBAC05, EBAC06, vzniknou následující řady:

ZEBOO1 P AEBOO5

AEBOO5 I ZEBOO1

ZEBOO1 P AEBOO6

AEBOO6 I ZEBOO1

Takto se zpracují všechny odkazy všech prvků A v knihovně zdrojových textů. Vytvořené věty se seřídí celým svým obsahem a vytvoří se dvouúrovňová tabulka. Nejvyšší úroveň obsahuje nejvyšší hodnoty klíčů a klíč přístupu k podřízené tabulce a má tedy tuto strukturu:

1 ZAKL

2 počet prvků

2 tabulka prvků

1 klíč /nejvyšší hodnota klíče na adresovaném místě/

3 adresa /například číslo stopy event. tabulky v paměti/

Druhá úroveň obsahuje vzestupně seříděné hodnoty klíčů:

1 SUB

2 počet prvků

2 tabulka prvků

3 klíč /13 znaků výše uvedené dvojice a směru/

V obou tabulkách se hledá metodou půlení intervalu. Nad touto dvouúrovňovou tabulkou musejí být realizovány operace vyhledávání /a to jak podle úplného tak i neúplného klíče/ a sekvenční /vzestupné či sestupné/ čtení prvků tabulky.

6. Tvorba seznamů, výskytů a odkazů prvků

Tato funkce slouží k rychlé orientaci v projektu. Realizuje se pomocí indexu zdrojové knihovny, který byl vytvořen v bodě 5. Manipulací a tímto indexem lze snadno získat odpověď na otázky typu:

- jaké soubory používají jednotlivé programy
- jaké programy jsou použity v jednotlivých úlohách, nebo naopak
- jaké programy používají daný soubor
- jaké úlohy používají daný program atd.

Tisk informačního systému může být zúžen např.:

- jaké programy skupiny úloh EF využívají soubory skupiny úloh EB
- jaké programy subsystému E využívají alespoň jeden formát obrazovky atd.

Další třídou možných dotazů tvoří dotazy typu:

- jaké programy nebo procedury jsou použity v jobech dané skupiny.

Poslední skupinu tvoří dotazy zřetězené; například:

- jaké programy jsou použity v jobech a v rámci těchto programů
jaké jsou použity datové soubory.

Formát výpisu informací by měl být dvojit:

a/ prvky jsou označeny pouze svými identifikátory

b/ prvky jsou označeny svými identifikátory a názvem /první řádek
z odstavce A/

Tento program je nutné vybavit vhodnými řídicími příkazy uváděnými
například v parametru programu.

a/ základem je identifikátor prvku, nebo jeho část např.:

EBZ005 - vybere se prvek EBZ005

EH - vyberou se všechny prvky začínající EH /skupina úloh/

b/ k základu je v závorkách připojit další parametry pro výběr,
pak je nutné za prvou závorkou uvést směr výběru /P = přímý
směr, I = inverzní směr/. Při zadání směru P se vyberou odkazy,
při zadání směru I výskyty prvků. Např.:

EBZ005/ /P - vyberou se všechny odkazy prvku EBZ005

E/ /P - vyberou se všechny odkazy všech prvků v subsystému

EBS/ /I - vyberou se všechny výskyty všech prvků začínající
znaky EBS.

c/ obsahem závorek lze řídit, které prvky se vyberou. Např.:

EBZ005/WS/P - vyberou se odkazy prvku EBZ005 /program/ na
všechny obrazovky /W/ a sestavy /S/, které tento
program používá.

d/ obsahem závorek lze též určovat výběr po zadaných úrovních. Např.

EBJ008/Z/AS//P - pro job EBJ008 se vyberou všechny programy /Z/,
které tento job využívá, a pro každý program
všechny soubory na disku /A/ a sestavy /S/,
které využívá.

e/ úroveň lze libovolně kombinovat. Např.:

EBU001 /X/J/Z/Q/Z//P

f/ za určení směru lze zadat znak T oddělený čárkou, což způsobí,
že ke každému tištěnému identifikátoru prvku se přiřadí jeho
název z prvního řádku odstavce A.

7. Řízený tisk dokumentace

Vhodnou strukturou řídicích příkazů tento program sestaví dokumentaci v požadovaném členění včetně obsahu. Předmětem dokumentace jsou členy knihovny zdrojových textů, součástí řídicích příkazů a odkazy, výskyty a seznamy prvků, tvořené dle bodu 6. Jelikož řídicích příkazů může být větší množství je vhodné aby byly uloženy v souboru. Tento soubor pak tvoří věty délky 80 bytů, kde první znak je řídicí a rozlišuje následující požadavky:

- a/ souhrnný tisk odstavců uvedeného prvku. Za řídicí znak se zapíše EBZA, což znamená, že se vytisknou všechny odstavce A všech programů, nebo EBSD, což způsobí, že se vytisknou vzory všech sestav.
- b/ uvedení celého identifikátoru prvku /bez řídicího znaku/ se vytiskne uvedený odstavec.
- c/ jiným řídicím znakem lze pak tisknout odkazy výskyty a seznamy dle bodu 6. Parametry v kapitole 6 následují po tomto znaku.
- d/ další řídicí znak uvede kapitoly za jejichž číslem po oddělovači se uvádí i název kapitoly.

Mimo to další řídicí znaky slouží pro editaci jako například vynechání daného počtu řádků, nová strana apod. Všechny vybrané odstavce se tisknou automaticky se svým názvem dle odstavce A daného prvku jak do dokumentace vlastní tak i do obsahu.

8. Závěr

Takto nebo obdobně navržený dokumentační systém, pokud má realizovány popsané funkce se stává významným pomocníkem všech analytiků, projektantů, programátorů a ostatních, kteří se zúčastní tvorby a realizaci projektu. Efektivní je to především v případě, že uváděný dokumentační systém je realizován v interaktivním prostředí a jeho uživatelé mají k němu současný a okamžitý přístup. Umožňuje všem zúčastněným informovat se o stavu prací, zavádět vytvářené prvky a odstavce, přehledy odkazů prvků a jejich vazeb. V konečné fázi pak umožňuje zavedené prvky vytisknout v požadované struktuře a tím získá dokumentaci v potřebném členění. Rovněž je jednoduchá jeho údržba /1 knihovna a 2 soubory/. Jeho realizace předpokládá vytvoření zhruba 9 programů nepřiližitě složitých.