

RECHENZENTRUM TH MÜNCHEN
ARBEITSGRUPPE BETRIEBSSYSTEME

INTERNSCHRIFT Nr. 22

THEMA: Dokumentation von Programmen

VERFASSTER: Sapper

DATUM: 22.4.1969

FORM DER ABFASSUNG

ENTWURF

☒ AUSARBEITUNG

ENDFORM

SACHLICHE VERBINDLICHKEIT

ALLGEMEINE INFORMATION
DISKUSSIONSGRUNDLAGE

☒ ERARBEITETER VORSCHLAG

VERBINDLICHE MITTEILUNG

VERALTET

ÄNDERUNGSZUSTAND

BEZUG AUF BISHERRIGE INTERNSCHRIFTEN

Vorkenntnisse aus:

Erweiterung von:

Ersatz für:

BEZUG AUF KÜNFTIGE INTERNSCHRIFTEN

Vorkenntnisse zu:

Erweiterung in:

Ersetzt durch:

ANDERWEITIGE LITERATUR

Zur Dokumentation von Programmen

Unterlagen für Teilnehmer am Programmierkurs der Arbeitsgruppe Betriebssysteme am Rechenzentrum der T.R. München; von Gerd R. Sapper, im April 1969 zusammengestellt.

Inhalt

1. Aufbau eines Programms
2. Dokumentation eines Programms
 - 2.1 Benutzeranleitung
 - 2.2 Programmbeschreibung
 - 2.3 Konstruktionsbeschreibung
 - 2.3.1 mit dem Quellencode
 - 2.3.2 anhand von Maschinenprotokollen
 - 2.3.3 unabhängig von Protokoll und Code
3. Zusammenfassung

1. Aufbau eines Programms

Ein Programm erscheint in folgenden Formen:

- (1) Entwurf (handschriftliche Aufzeichnungen des Verfassers)
- (2) Protokoll (für den Menschen lesbare maschinell angefertigte Listen)
- (3) Quellenprogramm (für den maschinellen Leser der Rechenanlage lesbarer Code)
- (4) Maschinencode (im Speicher des Betriebssystems abgelegte Befehle und Daten)

Die Arbeit an einem Programm beginnt mit bei (1) mit stichwortartigem Aufzählen, was das Programm im Einzelnen tun soll; grob strukturierten flußdiagrammartigen Ablaufplänen; evtl. Hinweisen auf die beim Codieren zu verwendende Programmierungstechnik. Daran schließt sich die Codierung an, die zunächst ebenfalls handschriftlich erfolgt. Bis zu diesem Punkt entstandene Unterlagen bilden einen wesentlichen Teil der Dokumentation. Bei Rechenanlagen, die die Anfertigung von Maschinenprotokollen nur in unübersichtlicher Form gestatten, bilden sie überhaupt die einzigen Unterlagen für eine Dokumentation.

2. Dokumentation eines Programms

Zur Dokumentation eines Programms gehören mehrere, in verschiedenen Stufen ausführliche, Beschreibungen. Ein nicht dokumentiertes Programm, das nur im Maschinencode vorliegt, ist wertlos. Dasselbe gilt fast genauso für nicht dokumentierten Quellencode, obwohl hier höhere Programmierungssprachen die Lesbarkeit eines Algorithmus wesentlich erleichtern.

Man benötigt

2.1 Die Benutzeranleitung oder Kurzbeschreibung

Sie wendet sich an den mit Vorkenntnissen ausgestatteten Benutzer, der dieses Programm lediglich anwenden möchte, keine extremen bzw. speziellen Anforderungen stellt und deswegen nur über die wichtigsten Steuerungsmöglichkeiten unterrichtet sein will. Auch soll eventuell einem potentiellen Benutzer schnell Auskunft gegeben werden, ob dieses Programm für ihn in Frage kommt.

Die Benutzeranleitung umfaßt deswegen nur wenige Schreibmaschinenseiten.

2.2 Die Programmbeschreibung

Ohne genau auf die Codierung einzugehen, liefert die Programmbeschreibung eine ausführliche Bedienungsanleitung, die dem Benutzer auch für Spezialanwendungen Auskunft gibt. Pro Tausend Befehle wird sie rund zehn Schreibmaschinenseiten umfassen.

Existieren diese beiden Teile der Dokumentation, so ist zwar eine Anwendung des Programms möglich, nicht jedoch eine Änderung oder das Verständnis von (wenn auch nur angeblichem) falschem Arbeiten. Hierzu benötigen wir die nachfolgend erläuterte Konstruktionsbeschreibung.

2.3 Die Konstruktionsbeschreibung

Man geht hierzu entweder vom Entwurf oder vom Maschinenprotokoll aus. Oft verwendet man zudem Programme, die vom Quellenprogramm ausgehend noch besser lesbare Listen erzeugen (z.B. spezielle Tabellierprogramme; Programme, die "Referenzen" drucken, alphabetisch geordnete "Adreßbücher" ausgeben usw.). Die so gewonnenen Protokolle werden ergänzt durch Informationen, die nicht maschinell erhalten werden konnten, im Einzelnen

Flußdiagramme; mathematische Formeln; Klammern und Pfeile, die einzelne Befehle verbinden; Tabellen, die die dynamische Belegung eines Freispeichers angeben usw.

2.3.1 Dokumentation im Quellencode

Obwohl es jede Programmiersprache gestattet, Kommentare in das Quellenprogramm aufzunehmen, darf man nicht annehmen, daß damit die Dokumentationsarbeit allein geleistet werden kann, denn Kommentare der oben genannten Art sind mit dem Zeichenvorrat eines Schnelldruckers schlecht darstellbar. Es läßt sich deswegen nicht umgehen, daß ein Maschinenprotokoll nachträglich mit ausführlichen handschriftlichen Kommentaren versehen wird und erst mit irgendwelchen Bildreproduktionstechniken ggf. vervielfältigte Exemplare hiervon als Dokumentation dienen. Bei der Zusammenstellung der Kommentare muß erhebliche Arbeit geleistet werden - durchschnittlich benötigt eine Maschinenanweisung eine Zeile für Erläuterungen, häufig jedoch mehr.

Punkte, die zur Zeit des Entwurfs des Programms sehr wesentlich und deswegen dokumentierungswert erschienen, erweisen sich später - wenn alles ausgetestet ist - als unwesentliche Nebensächlichkeiten; umgekehrt versteht ein unbefangener Betrachter gewisse Programmteile nicht, die der Verfasser pauschal und lapidar kommentierte, da er - mit dem Problem vertraut - sie für Trivialitäten hielt.

Bei in den Quellencode aufgenommenen Kommentaren ist eine nachträgliche Änderung nur mit maschinellen Hilfsmitteln möglich und unterbleibt oft wegen der Überlastung der hierzu nötigen Maschinen des Rechenzentrums. Bei der Maintenance wird deswegen oft zwar der fehlerhafte Befehl verbessert, die nun nicht mehr zutreffenden Kommentare bleiben stehen; auch unklare Kommentare werden nachträglich kaum so bereitwillig geändert, wie man das bei handgeschriebenen Erläuterungen tut.

In den Quellencode aufgenommene Kommentare werden überhaupt sehr unübersichtlich, wenn es sich um die Kommentierung einer Befehlsfolge handelt, die im Speicher erst dynamisch entsteht. Dies betrifft vor allem die Kommentierung eingriffsinvarianter Teile von Betriebssystemen, die an und für sich nicht eingriffssvariante Befehlsfolgen in einem Freispeicher aufbauen und dort durchlaufen.

Nicht zuletzt belastet auch jeder Kommentar im Quellencode den Übersetzungsvorgang am Rechner. Durch abgelochte Kommentare können nicht nur der Umfang des Quellencodes, sondern auch die Übersetzungszeit bis zum Faktor zehn steigen.

Man sollte sich deswegen beim Einfügen von Kommentaren beschränken auf

- (1) Namen von Programmen oder Programmteilen
- (2) Teile des Quellenprogramms, die der Übersetzer nicht protokolliert (in TEXAS z.B. Wortbestimmungen, GLOBAL/LOKAL-Anweisungen, Einsetzketten)
- (3) Allgemeine Bemerkungen, wie Datum der letzten Maintenance, Systemkennzeichen des Programms usw.

2.3.2 Dokumentation anhand von Protokollen

Besser wird das Quellenprogramm von speziellen Dokumentationsprogrammen, z.B. durch mit Protokoll ausgeführte Übersetzungen, zu Papier gebracht und dort handschriftlich kommentiert. Hierzu gehören:

- (1) Markieren von logischen Einschnitten durch von links nach rechts verlaufenden Trennungslinien (z.B. Anfang und statisches Ende von Unterprogrammen, zusammengehörenden Speicherbereichen usw.)
- (2) Markieren des dynamischen Endes von Unterprogrammen oder ähnlichen Programmteilen in einer einheitlichen Form.

- (3) Markieren von "Einsprungstellen" mit Angabe, woher auf diese Stelle gesprungen wird. Befinden sich Sprungziele in der nächsten Umgebung, so werden sie durch Pfeile am linken Rand verbunden (siehe Beispiel).

1. Beispiel

→(90)L13	DF	223	60	03DC	B3	T58	25	unterschiedliche Adr. → AC
UP Suchen Intervall	E0	224	A4	00EB	→S10	L16	-1	=0: Ende
	E1	225	05	1700	MU	MCE	0	
	E2	226	DC	0000	BAR	OU		Test obere Grenze nicht enthalten → L16
	E3	227	AF	00E5	→SG	L14		
	E4	228	36	00E8	→S	L16		
(91)L14	E5	229	05	1701	→MU	MCE	1	Test untere Grenze enthalten: → L15
	E6	230	8E	0000	BA	OU		
	E7	231	AE	00E9	→SK	L15		
	E8	232	36	00E8	→S	L16		
(91)L15	E9	233	04	7002	→EMU	B	2	Test drücken
	EA	234	38	000E	SU	14		
	EB	235	11	0004	→ZL	6		M/L 6: Intervall def.
(92)L16	EC	236	05	3603	→NU	S	3	Rg

Die senkrechten Teile dieser Pfeile legt man von vornherein verschieden weit vom Rand entfernt, so daß sie sich nicht überlagern können.

- (4) Zusammengehörende Befehlsfolgen werden entweder am rechten Rand mit einer Klammer versehen, oder - wenn es sich um den Aufruf eines globalen Unterprogramms, speziell eines Systemteils, handelt - eingerahmt.

2. Beispiel

SU	6
SYS	NE
C	B16
BH	B9
SN	F4
SYS	IAP
XBA	B11
XBA	B15
B2	B15
SU	11

Abfolge Per-Konstanten NEM

./ 7

- (5) Weiterhin sollen die einzelnen Befehle oder Befehlsgruppen erläutert werden, wozu sie hier dienen. Im obigen 2. Beispiel wäre die Bemerkung "Abfrage Positionszähler" allein wenig informativ, wichtig ist die Abkürzung NEM dahinter. Auch sollte man bei der Erläuterung bedingter Sprünge nicht allzu triviale Erläuterungen geben. Überflüssig wäre im 1. Beispiel eine Erläuterung von Sprungbedingungen nach Art der Befehlsliste ("Sprung nach ... wenn $AC \geq HR$ ") oder die Angabe, in welchem Maschinenregister sich die Grenzen befinden. Den Leser dieses Beispiels interessiert nur, daß der Befehl ZL6 ausgeführt wird, wenn $\langle T58 \rangle_3 = 0$ ist oder $\langle T58 \rangle_3$ in einem angegebenen Intervall liegt.
- (6) Bei den Aufrufen von Unterprogrammen, die Versorgungsparameter (insbesondere in Form von Sprungbefehlen) haben, trägt man relative Nummern von verwendeten Versorgungsparametern entsprechend dem dritten Beispiel neben der Befehlssequenz ein, dasselbe gilt für Sprungkaskaden oder ähnliche Sequenzen.

3. Beispiel

VEL	B18	+4
C	B18	+4
SU	L13	
0	B13	+1 $\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$ <i>Ende Progr.</i> <i>anf. Progr.</i> <i>"ASP"</i>
0	B13	
0	T61	
SU	L13	
XBA	ANF	-1 $\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$ <i>Ende Progr.</i> <i>anf. Progr.</i> <i>"10C"</i>
XBA	0	
0	T62	

B. 2:

00: DA Prog, DL (XBA) Rückblende ohne Blocken

LL: kein Rückblende nach diesem Plan (10C).

0	T20	0	prog. R.
0	T21	1	kein Befehl
0	T22	2	TK-AL
0	T23	3	anf. R.
0	T24	4	WZ-Überlauf
MA	T25	5	früherer
XBA	T26	6	ausl. Syn. Anfr.
XBA	T27	7	Überlauf
XBA	T28	8	Unterlauf
XBA	T29	9	ausl. BVP-Def.
XBA	T30	10	Schalt.
XBA	T31	11	ausl. BVP
XBA	T32	12	Bandende
XBA	T33	13	ausl. Syn. Anfr.
MA	T34	14	Pseudocharakter

15	MA	T35	Speicherplatz N bei falschem Wechsel
16	MA	T36	falsches Ende der EAS-Lsg.
17	MA	T37	falsche SE CAP-Endg.
18	XBA	T38	falsches Ende CAP-Adresse
19	XBA	T39	313 L&D: falsches R&L-Schild
20	XBA	T40	Alarm im System
21	XBA	T41	zu wenig CAP-Speicher
22	O	T42	Speicher
23	MA	T43	EA-Speicher nicht verfügbar
24	XBA	T44	POP

Auf diese Art wird klar gekennzeichnet, wo der Versorgungsblock zu Ende ist - bei Sprung-Kaskaden ist Ort und Art des Sprungziels mit einem Blick festzustellen.

2.3.3 Dokumentation unabhängig vom Protokoll oder Quellenprogramm

Die direkte Kommentierung eines Quellenprogramms ist dann nicht mehr sinnvoll, wenn der statische Aufbau des Programms kaum mehr den dynamischen Ablauf erkennen läßt. Wir denken hier nicht nur an die eingangs erwähnten eingriffsunempfindlichen Programme, die in einem ihnen dynamisch zugewiesenen Freispeicher Befehlssequenzen aufbauen und ausführen, sondern z.B. auch an matrixgesteuerte Übersetzungsprogramme, wo großer Wert auf eine übersichtliche Anordnung der Matrixfelder gelegt werden muß. In diesen Fällen entsteht die Dokumentation besser unabhängig vom Maschinenprogramm. Man kann zunächst die Dokumentation vom Maschinenprotokoll lösen, wenn man sich wie im 1. Beispiel das kommentierte Blatt an der eingezeichneten Trennungslinie durchschnitten und rechts mit den symbolischen TEXAS-Namen - speziell dem Namen des Unterprogramms selbst - ergänzt denkt.

Code und Dokumentation stehen dann auf zwei getrennten Blättern, die über die symbolischen Namen in der Assemblersprache verknüpft sind und getrennt gehandhabt werden können. Eine der-

artige Dokumentation empfiehlt sich, wenn die Codierung eines Problems noch sehr im Fluß ist und häufige Änderungen nötig sind, so daß ein Maschinenprotokoll schnell veraltet. Separat aufbewahrte Dokumentationsunterlagen ändern sich aber weniger schwerwiegend, bei der Korrektur von Codierfehlern z.B. gar nicht.

Bei dieser Methode bietet sich die Möglichkeit, logisch zusammengehörende Programmeinheiten - die in Wirklichkeit im Code an verschiedenen Stellen verstreut auftreten - auch zusammen zu dokumentieren; beispielsweise eine Liste von allen Indexspeichern, Merklichtern oder globalen Hilfszellen.

3. Zusammenfassung

Kommentare zu einem Programm sind um so unzugänglicher, je weniger hochentwickelt die verwendete Programmiersprache ist. Sie müssen mit der gleichen Sorgfalt angefertigt, verbessert und korrigiert werden, wie das Programm und seine Benutzeranleitung. Mit der Dokumentationsarbeit muß bereits beim Entwurf des betreffenden Programms begonnen und parallel zur Codierung weitergearbeitet werden. Die Dokumentation selbst sollte vom Quellencode und maschinellen Abzügen (Übersetzernlisten usw.) und damit von eventuellen Engpässen im Rechenzentrumsbetrieb (Locher, Tabelliermaschinen, Rechenzeit für Quellenbandänderungen) möglichst unabhängig sein.