

高等学校试用教材

COBOL程序设计

管理信息系统课程教材之二

哈尔滨工业大学 黄梯云 主编

GAO DENG XUE
XIAO JIAO CAI

机械工业出版社

高等学校试用教材

COBOL 程序设计

管理信息系统课程教材之二

哈尔滨工业大学 黄梯云 主编



机械工业出版社

JS195/12

本书内容包括: COBOL程序概述、COBOL数据部(Ⅰ)、
COBOL过程部(Ⅰ)、COBOL结构描述及标识部和设备部、
COBOL数据部(Ⅱ)、COBOL过程部(Ⅱ)、表处理、子程
序和排序、COBOL的文件处理及结构COBOL程序设计。

COBOL 程 序 设 计

(管理信息系统课程教材之二)

哈尔滨工业大学 黄梯云 主编

*

机械工业出版社出版(北京阜成门外百万庄南里一号)

(北京市书刊出版业营业许可证出字第117号)

机械工业出版社印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本 787×1092 1/16·印张9 3/4·字数232千字

1986年5月北京第一版·1986年5月北京第一次印刷

印数 00,001—13,000·定价1.70元

*

统一书号: 15033·6151

前 言

本书是根据1983年4月在杭州召开的机械工业管理工程类专业教材编审委员会第二次全体会议制定的教学计划和《管理信息系统》课程教学大纲编写的。管理信息系统课程教材分《管理信息系统导论(管理信息系统课程教材之一)》和《COBOL程序设计(管理信息系统课程教材之二)》两册编写。

COBOL语言是一种具有强大数据处理能力的计算机程序设计语言。目前,在国际上,已广泛地应用于商业及经营管理领域。本书的目的是使学员初步掌握COBOL程序设计方法,为把电子计算机这种先进工具应用于管理领域准备好一种基本的技能。

《COBOL程序设计》共分十章,包括COBOL程序概述、COBOL数据部(I)、COBOL过程部(I)、COBOL结构描述及标识部和设备部、COBOL数据部(II)、COBOL过程部(II)、表处理、子程序和排序、COBOL的文件处理和结构COBOL程序设计。

本书可作为高等院校管理工程专业的试用教材,也可供企、事业管理干部、电子计算机应用软件人员等作为参考书。还可作为在职干部培训教材。

本书第一、二、三章由浙江大学王燮臣编写;第四、五、六、七、八、九、十章由西安交通大学周金琪编写。哈尔滨工业大学黄梯云任主编。清华大学薛华成任主审。

参加本书审稿会的有:华中工学院邓汉源、北京工业学院刘振兴、大连工学院龙连文、陕西机械学院李枫、吉林工业大学杨杰、北京化纤学院王美华等同志,并对本书提出了许多宝贵意见。机械工业部教材编辑室对编写本书给予大力帮助,谨在此一并表示感谢。

由于编者水平所限,编写时间仓促,书中难免有不当之处,敬请读者指正。

编者

1984年6月

目 录

第一章 COBOL 程序概述	1	§ 5-5 带有 66 和 88 层号的数据描述	56
§ 1-1 COBOL 程序的特点	1	第六章 COBOL 过程部 (II)	60
§ 1-2 COBOL 程序的组成	1	§ 6-1 ACCEPT 语句和 DISPLAY	
§ 1-3 成绩程序的标识部和设备部	4	语句	61
§ 1-4 成绩程序的数据部	6	§ 6-2 GOTO 语句	63
§ 1-5 成绩程序的过程部	8	§ 6-3 ALTER 语句	65
第二章 COBOL 数据部 (I)	13	§ 6-4 PERFORM 语句和 EXIT 语句	66
§ 2-1 编写程序的初步工作	13	§ 6-5 条件和 IF 语句	71
§ 2-2 COBOL 的数据部	15	第七章 表处理	76
§ 2-3 COBOL 中数据项的初始化和		§ 7-1 表处理功能	76
VALUE 子句	18	§ 7-2 表和定义一个表	77
§ 2-4 COBOL 的数值编辑	19	§ 7-3 表元素的查找	82
第三章 COBOL 过程部 (I)	21	§ 7-4 表处理实例	86
§ 3-1 过程部的输入/输出动词	21	第八章 子程序和排序	90
§ 3-2 过程部的算术运算与数据传送		§ 8-1 COBOL 子程序	90
动词	23	§ 8-2 COBOL 排序	96
§ 3-3 过程部 IF 语句	27	第九章 COBOL 的文件处理	104
§ 3-4 过程部的次序控制语句	28	§ 9-1 文件和文件种类	104
§ 3-5 用伪指令的方法编制学生		§ 9-2 顺序文件的处理	105
成绩程序	30	§ 9-3 索引顺序文件的处理	110
§ 3-6 COBOL 程序的书写格式、		第十章 结构 COBOL 程序设计	117
编译和运行	35	§ 10-1 结构化程序设计原理	117
第四章 COBOL 结构描述及标识部和		§ 10-2 结构化程序设计方法	119
设备部	38	§ 10-3 结构化 COBOL 的模块描述	122
§ 4-1 COBOL 程序的结构	38	§ 10-4 结构化 COBOL 程序实例	124
§ 4-2 标识部和设备部	38	附录一 ANSI COBOL X3.23-1968	129
第五章 COBOL 数据部 (II)	43	附录二 ANSI COBOL 保留字表	130
§ 5-1 数据的表达方式	43	附录三 ANSI COBOL X3.23-1974	
§ 5-2 COBOL 的数据组织	45	的语言格式表	134
§ 5-3 PICTURE (字形) 子句	48	参考文献	149
§ 5-4 USAGE (用法) 子句	53		

第一章 COBOL 程序概述

§ 1-1 COBOL 程序的特点

COBOL, 全称 Common Business Oriented Language (面向管理的通用语言), 是一种最广泛地应用于企业经营管理领域的计算机程序设计语言。目前, 世界上计算机语言的发展日新月异, 各国科学研究人员研制了很多功能强大、使用方便的新的计算机语言, 如 PASCAL, C, PL/1 等语言。但迄今为止, COBOL 语言仍不失为应用最广泛的语言之一。

COBOL 最早的文本是 COBOL-60。1968 年美国标准化协会 (ANSI) 通过了 ANSI COBOL-68 文本。此后, 国际标准化组织 (ISO) 在 1972 年通过了 ISO COBOL-72 文本, 1978 年通过了 ISO COBOL-78 文本。近年来又出版了新的版本, 不断扩大其功能。

COBOL 最主要的特点是具有强大的数据处理能力。正因为如此, 它特别适用于商业、银行、服务行业、运输、工矿企业、政府机关等管理领域。

COBOL 是一种类似英语的程序设计语言。写一个程序如同写一篇用简单英文句子拼成的文章。因而这种语言容易看懂, 不必多作解释, 只要学过一点英语的人就能很快掌握它。

COBOL 也有缺点, 它的语句比较冗长、繁琐, 即使很简单的程序, 写起来也要面面俱到, 并且必须按严格规定的格式书写程序。COBOL 的另一缺点是科学计算能力较差。

COBOL 内容繁多, 要求教学过程由浅入深, 从最基本的语句着手, 逐步掌握较深入的内容, 因此, 本书把教学内容分为两大部分。一是基础部分, 它通一个具体的程序实例, 讲述 COBOL 最基本的构成和语句, 以便初步掌握 COBOL 程序设计的方法。第二部分则充实更多的语句, 使读者能够系统地掌握 COBOL 语言编写程序的技术。

学习 COBOL 语言要注意理论联系实际。可以说, 看许多条有关语句的具体规定, 还不如弄懂一个程序实例更能掌握要领。只有亲自编写程序, 并通过计算机这个最严格的老师考试上机通过程序, 才能真正学好 COBOL 程序设计。

§ 1-2 COBOL 程序的组成

每个 COBOL 程序由以下四个部 (DIVISION) 组成:

标识部 IDENTIFICATION DIVISION.

设备部 ENVIRONMENT DIVISION.

数据部 DATA DIVISION.

过程部 PROCEDURE DIVISION.

这四个部必须按上面的次序相继出现在程序中。各个部都有其特有的功能。

标识部 用于对程序命名。此外还可写一些备注性的内容, 如作者的姓名、编写与编译程序的日期等。

设备部 是程序中描述计算机硬件的部分。它指出使用的计算机主机和外围设备。

数据部 这是一个用于描述程序的数据结构。这些数据包括输入、输出及中间数据。

过程部 是最重要的一个部，它的功能是对数据进行处理及计算。

部以下可分成几个节 (Section)，但也可以不分节。部和节由段 (Paragraph) 组成。段由句子 (Sentence) 组成。句子由一个或几个语句 (Statement) 组成，一个句子以句 号结束。句子和语句由字 (Word) 组成。最后，字由字符 (Character) 组成。

COBOL 程序的组成 结构可概略地用图 1-1 表示。

表 1-1 是 COBOL 各组成部分的示例。

字是组成 COBOL 的基本单元。COBOL 的字由字母、数字与连字符组成。COBOL 字需符合的规定是：字符的个数最多不超过 30

个；至少有一个字母；连字符不能出现在字符组合的头与尾；字与字之间要用空格隔开。

COBOL 有两类字，保留字与用户字。保留字是具有专门语法意义的字，它要按 COBOL 语法规则来使用，不能作为用户字使用。例如保留字 ADD 只能出现在加法语句的开头位置。

COBOL 保留字又分为两类：一类叫基字；一类叫任选字。基字是在语句格式中必须书写的字，而任选字是在语句格式中可以省略的字。使用任选字的目的，仅是为了增加程序的可读性。

例如：读语句格式中：

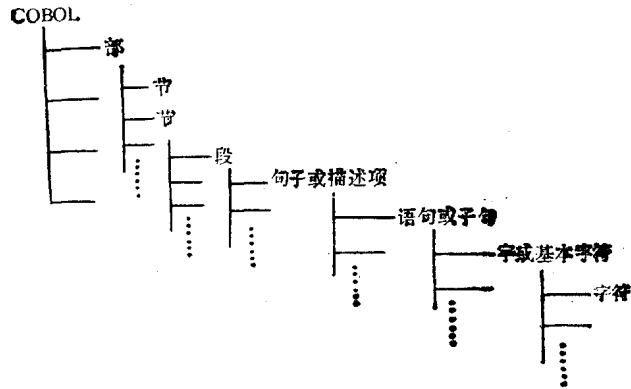


图1-1 COBOL的组成结构

表1-1 COBOL各组成部分示例

字	MOVE
语 句	MOVE 'Y' TO END-OF-FILE-FLAG
句 子	READ GRADE-CARD-FILE AT END MOVE 'Y' TO END-OF-FILE-FLAG.
段	READ-GRADE-RECORD. READ GRADE-CARDS-FILE AT END MOVE 'Y' TO END-OF-FILE-FLAG.
节	CONFIGURATION SECTION. SOURCE-COMPUTER. ... OBJECT-COMPUTER. ... SPECIAL-NAMES. ...
部	DATA DIVISION, FILE SECTION. ... WORKING-STORAGE SECTION. ...

READ	文件 名	RECORD	[INTO	数据 名]
	{ AT	END	强制语句	}
	INVALID	KEY	强制语句	

其中 RECORD, AT, KEY 是任选字，书写时可以省略。其他的是基字。

用户字是程序员自己定义的字，只要符合 COBOL 字组成的规定即可。用户字主要用来命名程序中的数据名（即变量名）、文件名、记录名以及过程部中的节名和段名，还可命名条件名（后面有专门介绍）。下面用表 1-2 举例说明 COBOL 的字：

COBOL 保留字举例：

ADD、READ、SECTION、MOVE 及 NOT 等。

用户字举例：

表1-2 COBOL 字举例

正 确 的 COBOL 字	非 COBOL 字
WANG 1	WANG □ LIN(是两个COBOL字)
XY	X·Y
MIS	MIS.
X-12	1-12
12X-1	12X·1

WANG-PROGRAM01、READ-CARD、PAY、HOURS 及 WORKS 等

字符是组成语句的符号，也是构成程序的最小元素。COBOL 源程序与 COBOL 字都由字符组成。

COBOL 字符集由下面 51 个字符组成，它包括五类字符，见表 1-3。

表1-3 COBOL 字符集

数字与英文字母	算术运算符	标点符号	关系运算符	货币号
0, 1, 2...9 A, B, C...Z	+(加) -(减) *(乘) /(除)	, (逗号) , (分号) .(句号) ' (引号) ((左括号)) (右括号) □(空字符)	>(大于) <(小于) =(等于)	\$

为了正确地描述 COBOL 语句格式，通常作如下习惯上的规定：

1. 凡保留字用大写英文字母，基字下面划一横线，无横线者为任选字。
2. 凡用汉字或用英文小字指明的语法单位是由程序员根据相应语法规则填写的语法单位。
3. 凡用方括号 [] 括起来的内容是任选的，其选用与否可由程序员决定。
4. 凡用花括号 { } 括起来的内容都是必须的。当有几个并列的语法成分出现时，应必选其中之一。
5. 省略号 “……” 表示其前的内容可以重复，若省略号跟在 [] 或 { } 后面，则表示该方括号或花括号中的整个说明项可以重复。
6. 格式中的逗号 “，” 和分号 “；” 都可以省略；如果写上它们，其后至少跟一个空格。为了强调该处要用空格，可以用一个空格符 “□” 表示。
7. 语法格式中的足码，只表示某同一语法单位的多次出现次数。

例：

<u>MOVE</u>	{ 标识符-1 义 字 }	TO 标识符-2 [标识符-3]……
-------------	------------------	--------------------

习 题

1. 下面名词中哪些是保留字？

PAYROLL LEFT WAIT WRITE RIGHT

TWENTY BRANCH MIDDLE ZERO TEST
 TRY CALL ANSWER OVERFLOW STOP
 GO CODE LOOK COMPUTE CALCULATE

2. 下列项目中有多少字?

PROCEDURE DIVISION.

INITIALIZATION.

OPEN INPUT TRANSACTION.

IF CONTROL-TEST-ON

THEN PERFORM MONTH-END.

3. 下列项目中有几个句子?

PERFORM READ-CARDS-ROUTINE UNTIL

EOF-FLAG IS EQUAL TO 'STOP'.

IF CUST-NUMBER IS EQUAL TO CUST-HOLD-AREA

THEN NEXT SENTENCE

ELSE MOVE DETAIL-LINE TO OUTPUT-AREA

PERFORM WRITE-ROUTINE.

PERFORM ACCUMULATE-ROUTINE.

4. 将下列COBOL术语从最小单元到最大单元按次序排列。

SECTION PARAGRAPH SENTENCE WORD

DIVISION STATEMENT

§ 1-3 成绩程序的标识部和设备部

学习 COBOL 的最好方法是具体编写程序,因此,下面以学生成绩程序为例介绍 COBOL 程序的最基本内容。

学生成绩程序要求打印出一张学生的成绩报表(参见表 1-4),其内容包括每个学生的学生号和成绩两项信息。报表的末行打印全班中 A 等成绩的总人数。图 1-2 是它的系统流程图。由图可知,该程序的输入是每个学生的一张输入卡片,卡片上穿的数据是学生号与该学生的成绩。

下面是成绩程序的第一个部,叫做标识部:

IDENTIFICATION DIVISION.

PROGRAM-ID. GRADEDS.

AUTHOR. K. DOLAN.

标识部中程序员首先给程序命名,其次可写上某些备注,如著者姓名、程序编写日期等。上述语句中 IDENTIFICATION DIVISION、PROGRAM-ID 与 AUTHOR 都是保留字,而 GRADES 与 K. DOLAN. 是由程序员确定的用户字。程序名不长于 8 个字符。(CRO-MEMCO 系统程序名不长于 6 个字符),但著者姓名的长度不限。

第二个部是设备部。

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM 370.

表1-4 成绩程序打印输出表

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
--	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

字。其他全是保留字。还需说明两个设备的名称 SYSIN 与 PRINTER。SYSIN 即系统的输入设备, PRINTER 即打印机。这两个字不是 COBOL 语法所规定的字, 它是根据特定的计算机系统规定的该系统的外部设备名, 如西门子 7738 系统的读卡机命名为 SYSIN, 宽行打印机命名为 PRINTER。不同的机器系统有它特定的设备名称。编写程序时只要查阅机器的有关手册, 就可确定输入-输出设备的名称。

习 题

1. 设备部的哪一节识别计算机系统?
2. 设备部的哪一节命名输入与输出文件?
3. 下列 COBOL 项目中哪些是文件名?

```
SELECT ACCTS-RECEIVABLE-MAST ASSIGN TO DA-2314-S-MAST.
SELECT PRINT-REPORT ASSIGN TO UR-1403-S.
```

4. 写出一个 COBOL 段, 用来命名一个名为 UPDATE 的程序。

§ 1-4 成绩程序的数据部

成绩程序中的数据, 都必须在数据部定义和描述。COBOL 是用文件的形式来描述数据的。上述 GRADE-CARDS-FILE 是输入文件, GRADE-REPORT-FILE 是输出文件。

输入文件 GRADE-CARDS-FILE 由一叠成绩卡片上的数据组成。

由程序产生的打印输出也以文件的形式描述, 称为打印输出文件, 文件名即 GRADE-REPORT-FILE, 每次打印一行, 就是输出文件的一个记录。

成绩程序的数据部描述如下:

```
DATA DIVISION.
FILE SECTION.
FD GRADE-CARDS-FILE
  LABEL RECORDS ARE OMITTED
  RECORD CONTAINS 80 CHARACTERS.
01 GRADE-CARD-RECORD...
FD GRADE-REPORT-FILE
  LABEL RECORDS ARE OMITTED
  RECORD CONTAINS 133 CHARACTERS.
01 GRADE-REPORT-RECORD.
  05 FILLER...
  05 PRINT-LINE...
```

数据部的第一节是文件节 FILE SECTION。在设备部的 SELECT 语句中所命名的每一个文件必须在文件节中更完整地描述。字母 FD 是 FILE DESCRIPTION 的缩写, 它标明了这是有关该文件的描述项。文件名紧接在 FD 的后面。LABEL RECORD 子句 (子句是语句中的一个短句 Clause) 指出文件的特定标记。如卡片文件的标记可以省略, 写作

“LABEL RECORD IS OMITTED”，因为它不需要特定的标记，而对磁盘文件则应写上“LABEL RECORD IS STANDARD”。它说明了文件的标记是标准的。计算机会自动地确定文件的标记。

下一个子句是 RECORD CONTAINS 80 CHARACTERS。它指出记录的长度由80个字符组成。卡片的规格是80列，因而记录以80个字符为限。整个FD句子的结束处有一个句号。

FD描述体要求COBOL的编译程序在计算机的储存中留出一个储存空间，称为缓冲区(Buffer)。缓冲区用于计算机的输入与输出，也就是数据可以写入缓冲区，也可以从缓冲区读出。程序员往往频繁地将输入缓冲区的数据，复制到一个中间的工作区，然后进行有关的数据处理过程。经过处理的数据，再由工作区复制到输出缓冲区，然后通过输出设备将数据输出(参见图1-3)。

RECORD CONTAINS 133 CHARACTERS子句指出每个打印记录有133个字符。其中第一个字符用于控制打印机走纸的信息。而其他132个字符是打印的字符。它们都应在输出文件中说明。

第二节是工作存储节 WORKING-STORAGE SECTION。

学生成绩程序的工作存储节举例如下：

WORKING-STORAGE SECTION.

77 NUMBER-OF-A-GRADES...

01 HEADER-LINE.

05 FILLER...

05 FILIER...

05 EILIER...

01 GRADE-LINE.

05 FILLER...

05 STUDENT-ID-OUT...

05 FILLER...

05 GRADE-OUT...

05 FILLER...

NUMBER-OF-A-GRADES是累加A成绩的数据项。在成绩处理中，凡是A等成绩都在该数据上累加。它是一个独立的数据项，不再细分。

HEADER-LINE设置了报表头中需打印的字符。

GRADE-LINE组成了一个输出报表的一行，包括学生号与学生成绩。

GRADE-CARD-WORK-AREA是一个工作区，当每个输入记录读入计算机进行处理之前，应把该记录复制到一个工作区，它共有三项，STUDENT-IN，GRADE-IN，与FILLER。FILLER是一个保留字，它在数据描述中作为一个不需专门命名的数据项，我们称之为充填项，它在程序中经常使用，特别是在工作储存节中经常使用。

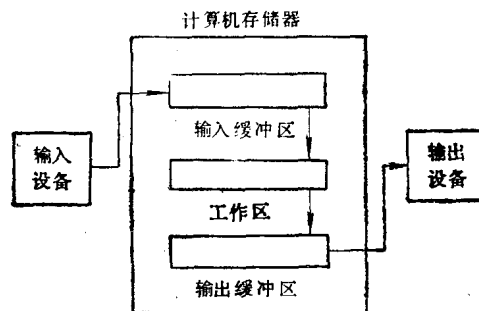


图1-3 计算机中数据的传送

01, 05 是数据项的层号。01 是记录的最高层, 它定义了一个数据集合, 即一个记录, 02 到 49 层号用于记录中所属的各数据项。

77 层是工作存储节中的一个特殊层号。在工作存贮中某些无层次的项, 它们不需要细分, 可定义为一个独立的数据项。给予一个专门的层号, 77 层。

习 题

1. 指出数据部两个节的名字。它们的功能是什么?
2. 假设建立一个叫做EMPLOYEE-NUMBER的数据项, 并在过程部中以 EMPLOYEE-NO 的名来访问它。计算机能否知道所指的是同一个项目? 为什么是或为什么不是?
3. 给一个名为CHAGE-ACCT-TRANSACTIONS的卡片文件编写文件描述项(FD)。
4. 在下面COBOL项中, 哪些是文件名? 哪些是记录名? 哪些是数据名?

FD PAYROLL

 LABEL RECORD ARE OMITTED
 RECORD CONTAINS 80 CHARACTERS.

01 PAYROLL-INFO PIC X(80).

FD PRINTOUT

 LABEL RECORD ARE OMITTED
 RECORD CONTAINS 133 CHARACTERS

01 PRINT-LINE.

03 VERTICAL-CONTROL PIC 9.

03 OUTPUT-AREA PIC X(132).

§ 1-5 成绩程序的过程部

编制过程部的方法很多, 主要的方法有两种。一是绘制流程图的方法, 另一种是编写伪指令(Pseudocode)的方法。下面先介绍绘制程序流程图编写成绩程序过程部的方法。关于编写伪指令的方法将在后面第§ 3-5中介绍。学生成绩程序的程序流程图如图1-4所示。

学生成绩程序的过程部分为五个段:

1. 起始段;
2. 打报表头段;
3. 处理段;
4. 总计段;
5. 工作结束段。

这五个程序段的内容和上述程序流程图是一致的。第一段是启动程序打开文件。第二段是打印报表的表头。第三段是处理一张输入卡片, 判别该学生的成绩是否A等, 若是A等, 则将其计数, 然后打印该学生的成绩。每处理一张输入卡片, 重复地执行该处理段的处理过程, 一直到处理完所有的输入卡片为止。第四段是打印A成绩总数的打印行。第五段是任务结束段, 内容包括关闭文件和程序停止运行。

整个学生成绩程序的源程序如下:

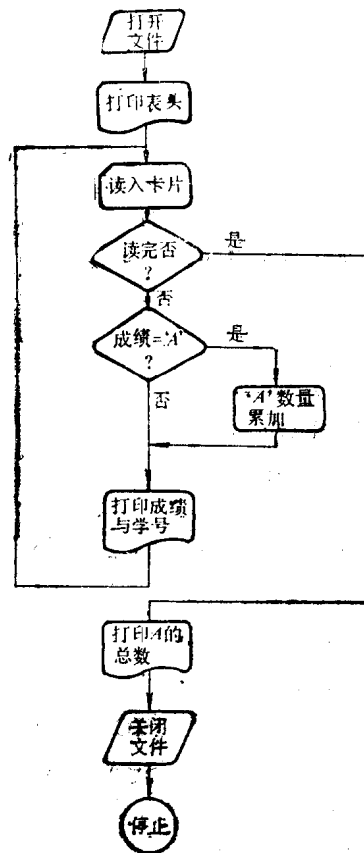


图1-4 学生成绩程序的程序流程图

IDENTIFICATION DIVISION.

PROGRAM-ID. GRADES.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT GRADE-CARDS-FILE ASSIGN TO SYSIN.

SELECT GRADE-REPORT-FILE ASSIGN TO PRINTER.

DATA DIVISION.

FILE SECTION.

FD GRADE-CARDS-FILE LABEL RECORDS ARE OMITTED
RECORD CONTAINS 80 CHARACTERS.

01 GRADE-CARD-RECORD PIC X(80).

FD GRADE-REPORT-FILE LABEL RECORDS ARE OMITTED
RECORD CONTAINS 133 CHARACTERS.

01 GRADE-REPORT-RECORD.

05 FILLER PIC X.
 05 PRINT-LINE PIC X(132).

WORKING-STORAGE SECTION.

77 NUMBER-OF-A-GRADES PIC 999 VALUE 0.

01 HEADER-LINE.

05 FILLER PIC X(10) VALUE SPACES.
 05 FILLER PIC X(6) VALUE 'GRADES'.
 05 FILLER PIC X(116) VALUE SPACES.

01 GRADE-LINE.

05 FILLER PIC X(5) VALUE SPACES.
 05 STUDENT-ID-OUT PIC X(10).
 05 FILLER PIC XXX VALUE SPACES.
 05 GRADE-OUT PIC X.
 05 FILLER PIC X(113) VALUE SPACES.

01 TOTAL-LINE.

05 FILLER PIC X(22) VALUE
 'NUMBER OF A GRADES IS'.
 05 NUMBER-OF-A-GRADES-OUT PIC ZZ9.
 05 FILLER PIC X(107) VALUE SPACES.

01 GRADE-CARD-WORK-AREA.

05 STUDENT-ID-IN PIC X(10).
 05 GRADE-IN PIC X.
 05 FILLER PIC X(69).

PROCEDURE DIVISION.

A1. OPEN INPUT GRADE-CARDS-FILE.

OPEN OUTPUT GRADE-REPORT-FILE.

A2. MOVE HEADER-LINE TO PRINT-LINE.

WRITE GRADE-REPORT-RECORD AFTER
 ADVANCING TOP-OF-PAGE.

PROCESS.

READ GRADE-RECORDS-FILE INTO GRADE-CARD-WORK-AREA
 AT END GO TO TOTAL.

IF GRADE-IN EQUAL TO 'A'

ADD 1 TO NUMBER-OF-A-GRADES.

MOVE STUDENT-ID-IN TO STUDENT-ID-OUT.

MOVE GRADE-IN TO GRADE-OUT.

MOVE GRADE-LINE TO PRINT-LINE.

WRITE GRADE-REPORT-RECORD AFTER

ADVANCING 2 LINES GO TO PROCESS.

TOTAL.

MOVE NUMBER-OF-A-GRADES TO NUMBER-OF-A-GRADES-
OUT.

MOVE TOTAL-LINE TO PRINT-LINE.

WRITE GRADE-REPORT-RECORD AFTER
ADVANCING 3 LINES.

JOB-END.

CLOSE GRADE-CARDS-FILE GRADE-REPORT-FILE.

STOP RUN.

现在对照这个源程序，对过程部作概要的说明：

第一段，起动段是二个 OPEN 语句，它分别打开了 GRADE-CARDS-FILE 与 GRADE-REPORT-FILE 两个文件。

第二段，打印报表头段。第一语句是 MOVE 语句。它将已安排好的数据项 HEADER-LINE 传送到另一个数据项 PRINT-LINE 中，这样就形成了打印报表头的行。下一句是 WRITE 语句，意为在打印机上打印一个打印的记录，该记录名是 GRADE-REPORT-RECORD。AFTER ADVANCING TOP-OF-PAGE. 即要求输出报表从一张打印纸的页头开始打印，也就是另起一页打印。

第三段处理输入数据的程序段。首先是 READ 语句。它要求计算机从输入文件中取得一个记录。这是读入一张卡片记录，把它写在计算机内存的输入区。INTO GRADE-CARD-WORK-AREA 是由计算机将输入记录的数据复制到数据项 GRADE-CARD-WORK-AREA 上。AT END 是由计算机去判断被读的记录，是数据记录，还是文件的结束记号，如果是文件的结束记号，则执行 AT END 后面的语句 GO TO TOTAL，即要求程序的执行次序转移到名为 TOTAL 的程序段去。程序从该段的第一条语句开始往下运行。若 READ 语句读到的记录是一个数据记录，则执行 READ 语句的下一条语句，即 IF 语句。

IF GRADE-IN IS EQUAT TO 'A' THAN
ADD 1 TO NUMBER-OF-A-GRADES.

这是一个条件语句，它将数据项 GRADE-IN 的值与 A 作比较，GRADE-IN 的值是否等于 A，这就是条件语句的一个条件。若该条件为真则执行 THAN 以后的强制语句，若条件为假，则执行 IF 语句后的下一条语句。

上述语句中应注意有两项数据 1 与 'A'，1 与 'A' 都称为文字 (Literal)，1 是一个数值文字，而引号内的 'A' 是一个非数值文字。文字都具有值。数值文字是一个数值，而非数值文字则是一串在引号中的字符串。

后面是三条 MOVE 语句与一条 WRITE 语句。三条 MOVE 语句是将有关输入的数据传送到成绩报表记录中的有关数据项中去，并形成了一个学生成绩的打印行的数据。WRITE 语句是打印一个成绩报表的记录。AFTER ADVANCING 2 LINES, 意为打印时空一行打印一行。该段的最后是一个 GO TO PROCESS 语句。意为程序运行次序转到 PROCESS 段。即从 PROCESS 段的第一条语句开始运行。

通过 GO TO PROCESS 语句，该程序重复执行 PROCESS 程序段，也就是开始读第二张卡片，并重复执行全部 PROCESS 程序段的语句。这样重复地进行这个处理循环，依

次把所有的输入卡片全部处理完毕。最后程序将读到该输入卡片中的文件结束记号。文件结束记号也穿在一张卡片上。当 READ 语句读到文件结束记号时则执行 GO TO TOTAL 语句，程序的运行次序即转移到 TOTAL 程序段。

第四段 TOTAL 程序段是建立一个统计 A 成绩总人数的打印行。它也由两个 MOVE 语句建立打印行的数据，由 WRITE 语句在报表之末打印一行。至此整个程序要进行的工作已经做完，即打印出一张符合设计要求的 student 成绩报表。

第五段 JOB-END 是一个程序结束段。程序最后要求把所有的文件关闭，并停止运行。这里使用了 CLOSE 与 STOP RUN 语句。