Chapter 5

Technical description of the system

5.1. Installation

The main processor of the system is an LSI 11/23 with 256Kb of memory. The main disc is at present a 20Mb Winchester configured as 8 RK05 discs. The system runs as a multi-user system under Unix version 7. All the programs are written in C.

The X25 front-end consists of an additional 11/02 processor and some interface boards, running a program on an EPROM chip, and in addition some programs installed on the main system. The programs and the EPROM were obtained from York University. It should be noted that it is now possible to get a cheaper front-end system based on a microprocessor.

5.2. Source code

The source code for **cirt** is contained in five files. Broadly, the various functions are divided among the files as follows:

cirt.c contains the main calling program, interpretation of the cirt command language, initiating and terminating routines etc.

search.c contains the function for converting the weighted search into a series of Boolean searches.

print.c contains routines for displaying retrieved documents, collecting relevance judgements etc.

lex.yy.c contains routines for interpreting communications from the host system.

x29.c contains routines for interfacing with the network.

Following is a more detailed description of these files and the various functions they contain. Further files, not listed here, contain definitions for the C programs.

5.2.1. Cirt.c

This file contains **main**, which performs some initialisation steps and then switches control according to the user's command. Main calls **ofname**, which generates a temporary output file name, and **get_com**, which gets a command from the user and identifies it. Get_com uses four more functions: **argfree, match_pos, pmatch** and **rmatch.**

Depending on the particular user command, main may also call **login** (which performs the automatic login); **termin**, which checks whether a term is already in the query; **delqt**, which deletes a term if it has not

already been searched; **liststr**, a recursive function which displays appropriate nodes of the search tree; **newwts**, which calculates new weights for the search terms given relevance data (and itself calls **newstw**, a recursive function which updates the node weights in the search tree); **reset**, which clears the query and search tree; and **quit**, which closes the network connection and exits. Main may additionally call **ok**, which asks the user if they want to proceed, and **aquit**, which closes the network connection and aborts.

Additional routines called by main and some of the other functions mentioned appear in the other files listed below.

5.2.2. Search.c

Recursive routines are indicated by (R).

The main search routine (as described in chapter 4) is **rsearch** (R); it is called by main, and calls **rmwt** (R) (also described in chapter 4). Rsearch also calls **search**, which submits a search statement to the host and returns the results. Also called by main are: **nqlist**, which adds a new query term to the list, and calls search and **getrels** (which asks the user to input the number of relevant documents for this term); **tdytree**, which prepares the search tree for a new search, and calls search and **scntree** (R); **limit**, which sets up a limiting query, and calls search (not yet fully implemented, as discussed in chapter 6).

This file also contains two routines called by reset (cirt.c): **clrqry** (R) and **clrtree** (R), which clear the query and the tree respectively. Search is also called by orrels in print.c, and calls yylex (lex.yy.c) and some functions in x29.c.

5.2.3. Print.c

The main printing routine is **dprint**, which controls the retrieval of document descriptions from the host and their display on the terminal. Dprint calls: **bnode** (R), which finds the highest-weighted node in the tree which has some documents unseen by the user, and calls **bestf**; **freedoc** (R), which clears the document; **dseen**, which checks whether a document has been seen by the user; **verdict**, which asks for the user's response to the document; **dinsert**, which adds a document to the list of seen documents; **addrels**, which adds to the relevant document counts for each term; and **prdoc**, which adds a document to the print file.

This file also contains **orrels**, which does a search on the identifiers of the relevant documents, is called by main (cirt.c) and calls search (search.c); **dsclear** (R), which clears the list of seen documents, and is called by reset (cirt.c). Freedoc is also called by yylex (lex.yy.c). Dprint also calls yylex (lex.yy.c) and functions from x29.c.

5.2.4. Lex.yy.c

There is a Unix facility called LEX and described as a "Lexical Analyzer Generator", which generates C routines to be used for simple

lexical analysis of text. We have used LEX to generate a routine to analyse the responses received from Data-Star. The principal function in this file is **yylex**, which is called by login (cirt.c), search (search.c) and dprint (print.c). Yylex calls some more functions within lex.yy.c; also freedoc (search.c) and one function from x29.c.

5.2.5. X29.c

This file contains a large number of function with a complex network of calls. These functions handle all the direct communication with the network, and make use of the York softare. Functions in this file are called by main, login, quit and aquit (cirt.c); search (search.c); dprint (print.c); and yylex (lex.yy.c)