

RC International

RC5000 File System

File Processor

Operators Guide

Keywords:

RC5000, File System, Winchester disc, FP, File Processor.

Abstract:

This manual describes the File Processor and the file utility programs in the RC5000 system.

Date:

December 1991

Author:

Henning Jakobsen

PN: 99001411

Copyright

Copyright • 1991 RC International (Regnecentralen a/s) A/S Reg.no. 62 420

All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without the prior written permission of RC International, Lautrupbjerg 1, DK-2750 Ballerup, Denmark.

Disclaimer

RC International makes no representations or warranties with respect to the contents of this publication and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, RC International reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of RC International to notify any person of such revision or changes.

PN: 99001411

1. INTRODUCTION

Purpose

The File Processor and the file utilities are the components that constitutes a small disk operating system controlled from the local console based on the File System used in the RC5000 system.

Prerequisites

The RC5000 Operating System (SW50020) and the File System (SW50220) must be loaded and started before the File Processor is started.

Operation

The File Processor is started by giving the command "run fp" to opsys.
The file system must be running, i.e. the process "fs" may be seen by a "list fs" to opsys.

The fp writes a prompt character (\$) as a ready for command sign.

2. LOGIN PROCEDURE

Type "login" followed by the pathname to a directory. The mentioned directory becomes current directory. The root directory is named "."

The fp writes:

"Welcome to the File System <current date from opsys>"

If the date is 1900.00.00 then you are recommended to set the proper date by the "date" command to opsys, e.g. date 91.12.13 8.15.00.

Example: login pax7cat

3. FILE PROCESSOR COMMANDS

help Writes list of commands and parameters. The capital letters shown should be used to start the command in question. the word `pathname_pattern` means that a `*` may be used as a wildcard character for multi file operation.

run <pathname> The given file is loaded and started. Fp waits for the program to stop, i.e. the program calls the routine `fp_halt`.

exit Close current directory and be ready for a new login. If fp is the only user of the file system, the disk is powered down if possible.
Alternative command: `logo`

cd <pathname> Change directory to the specified directory.
Alternative command: `cu`

Example: `cd /autoloadcat`

rm <pathname_pattern> Delete all files which matches the specified `pathname` pattern. You are prompted before deletion of each file. Answer to the prompt must be y or n.
Alternative command: `del`

Example: `rm olds*`

ls <pathname_pattern>

Write a list of all files which matches the specified pathname pattern.

mysub	1991.04.16:08:10	10240	rwsda rwsda rwsda	catalog
myfile	1991.04.16:14:10	34711	rwsda rw--a r----	data
myinx	1991.04.16:14:12	1280	rwsda rw--a rw---	data

The list shows:

<filename> <last update> <size> <access rigths> <type>

filename: up to 16 characters.

last update: date for last write into the file.

size: amount of bytes in the file.

access rigths: for authorization, for link, for access:

r: read allowed

w: write allowed

s: size of file may be changed

d: deletion is allowed

a: access rigths may be changed

type: catalog (= directory) or data (= not directory)

Alternative command: cat

loa <pathname>

Transfer (load) the specified file from disk to memory and link program modules with existing modules. The loaded modules may later be started by a run command.

loo <pathname>

Lookup the specified file and write same line as ls. If the file is open fp writes: ** fileerror rejected or open.

mk <pathname>

Make a new directory.

cp <source pathname> <destination pathname>

Copy the specified source file into a new file. If destination file allready exists you will be prompted before the copy is made.

Alternative command: cop

ln <original pathname> <new pathname>

Link a file to an existing file (alias name).

mv <old pathname> <new pathname>

Rename (exchange the name of) a file.

Alternative command: re

pro <pathname> <a/i/p> rwsda
Change protections (access rigths) on a file.
a for authorization,
i for inheritable by link,
p for access protection
r: read allowed
w: write allowed
s: size of file may be changed
d: deletion is allowed
a: access rigths may be changed

Example: pro myfile i rs

len <pathname>
Used to show or change the length in bytes of a file.
Fp asks for a new length. If no new value is typed then
the length is unchanged.

size <stacksize>
Change the default stacksize (in words) used by opsys to
create a process.

test <on/off>
Change a test flag in fp. Used to get some auxiliary output
from fp.

un <programname>
Let opsys unload the specified program.

4. FILE PROCESSOR ROUTINES

4.1 Halt

Declaration: Procedure fp_halt (trace: boolean);

Function: Used from a program running under fp to stop the program itself and let fp continue. If trace is true, the stack of the caller is written before fp takes over.

4.2 Replace

Declaration: Procedure fp_replace (var commandline: ! fp_line);

Function: Used from a program running under fp to stop the program itself and let fp continue with the command given in the commandline.

4.3 System

Declaration: Function fp_system (paramno: integer;
var num: integer; var name: fs_identifier): integer

Function: Used from a program running under fp to get access to its program call line.

paramno: The wanted parameter number. Counting from zero.

num: The numeric value of the parameter.

name: The parameter as text, up to 16 characters.

Return value in fp_system: separator type * 256 + paramtype

Separator types:

NL	0
NUL	2
SP	4
=	6
.	8

Parametertypes:

empty	0
number	4
name	10
"text"	18

4.4 Odd

Declaration: Function odd (p: integer): boolean;

Function: May be called from any program to get least significant bit of an integer. Returned value: (p and 1) = 1 .

5. UTILITY PROGRAMS

5.1 COMPARE

Writes the differences between to files.

Program call:

(out =) compare <file1> <file2> (first.pageno) (last.pageno) (maxdifs.words)

Default values are: out= console, first= 0, last= 32767, maxdifs= 32767

5.2 CORRECT

Updates single byte positions of a file.

Program call:

destination = correct <source>.<pageno> (addr.<byteno> <correction>

<correction>::= <if part> <then part>

<if part>::= if <old value> / ifch <old character>

<then part>::= then <new value> / thench < new character>

5.3 FSFORMAT

Initializes the disk with a new file system, or check disk.

For further details see PN 99001385: FSFORMAT, Operators Guide.

5.4 FTS

Transfers files via LAN to or from a remote LAN server.

Requirements: The IMC system must be running.

Program call:

fts.<rr/wr> (s.<server>) (u.user)) (<options>) (f.<remotefile>) (localfile))

<options>:

p.lp	for print on printer lp
t.yes	for transfer of ASCII text files
n.yes	for no-overwrite of destination file
q.yes	for quiet operation (only errors shown)
w.<password>	for denotion of password
l.<lanno>	for use of alternative IMC.
c."<UNIXcommand>"	for remote execution of a UNIX command, up to 16 characters long

5.5 GETFPA

Transfers a file from RC8000 via a FPA100.

Program call:

<destination> = getfpa

5.6 LIST

Writes the contents of a text file (or some lines thereof).

(out =) list <source> (first.firstline) (last.lastline) (span.<intervals>)

<intervals>::= .firstline.lastline

Default values: out= console, firstline= 0, lastline= 32767

5.7 PLIB

PLIB is a utility program for manipulation of RTP object programs. The program may be called in different modes: list contents of a library, list contents of a coreimage, insert or exchange modules in a library, extract modules from a library, or delete modules from a library.

Program call:

(<destination> =) plib.<mode> (objectfiles/modulenames) (lib.<source>)

<mode>::= lookup / image / insert / extract / delete

Examples:

plib.lookup imcplib	list contents of objectfile on console.
clst=plib.image boot35027	list an absolute coreimage file.
	(clst may be printed by a fts.wr command)
new=plib.insert bwld lib.old	
	file bwld is inserted into old.
imc2=plib.delete log9330 lib.imcplib	
	removes log9330 from imcplib.

5.8 PRINT

Writes the contents of a file as hexadecimal bytes and ASCII characters.

Program call:

(output =) print <source> (first.firstpageno) (last.lastpageno)

firstpageno to lastpageno denotes an interval to be shown.

Default output is console.

5.9 SHOW

Writes a survey over the file system format and status.

Program call: show or show all

Output on console:

Volume: <volumename> init: <date of initialization>

slicesize: <pages> total slices: <slices> bad: <no of bad slices>

free: <no of free slices> at <no of free areas> locations

If the parameter "all" is present the the slice allocation table is shown too. The table is shown as one digit per 8 slices. The digit denotes amount of free slices. Zero is shown as a dot "."

5.10 TYPEIN

May be used for primitive data entry of text to a text file.

Ctrl-y terminates the input.

Program call:

(<destination> =) typein

Default value: destination= console