
RC9000-10/RC8000

SW8585 Compiler Collection

ALGOL Coroutine System

Testudskrivning

RC Computer

Keywords:

RC9000-10, RC8000, ALGOL, Coroutines

Abstract:

Dette skrift beskriver en metode til udskrivning af testfiler genereret af ALGOL Coroutine systemet.

Date:

Januar 1989

PN: 991 11291

Copyright © 1988, Regnecentralen a.s./RC Computer a.s
Printed by Regnecentralen a.s, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

Indholdsfortegnelse

1. Introduktion.....	1
2. Metoden.....	2
2.1 Proceduren test_print.....	2
2.1.1 Kald.....	3
2.1.2 Fejludskrifter.....	4
2.2 Proceduren print_record.....	4
2.2.1 Eksempel på print_record procedure.....	5
2.3 Program.....	6
Appendix A. Referencer.....	8



41.

112

65

1. Introduktion

Dette skrift beskriver en metode til fleksibel udskrivning af testfiler genereret af RC8000 Coroutine System. Brugervejledningen til coroutinesystemet, ref. 1, angiver at filen kan udskrives med fp-kommandoen

```
print <fil> word words.8
```

men det resulterer i en ren numerisk udskrift, der er meget besværlig at fortolke. Formålet med den udskrivningsmetode, der beskrives heri, er at lette fortolkningen af testoutput samt at muliggøre udvælgelse af individer på en nem måde.

Metoden er første gang anvendt i forbindelse med Scroll Mode Mapping (SMM) modulet i CENTERNET, hvilket er grunden til at den individuelle del af metoden er illustreret med det, der anvendes til SMM testoutput udskrivning.

Teksterne til procedurerne `test_print` og `print_record` findes i henholdsvis bilag B og C. Manualen igennem anvendes termen `bs-fil` om begrebet baggrundslager fil (engelsk: backing storage file).

2. Metoden

Metoden består af en standarddel og en brugerafhængig del.

Standarddelen sørger ud fra en passende trimning for at

- positionere i filen til først ønskede individ
- læse individerne
- udvælge de individer, der skal udskrives
- afslutte læsningen.

Den brugerafhængige del kan betragtes på to forskellige niveauer, den faste del og den varierende del. Den faste del er den udskrivningsmetode, der skal anvendes på de udvalgte individer, mens den varierende del er den trimning, der er nødvendig for i hvert enkelt tilfælde at få positioneret filen, samt at få udvalgt de ønskede individer fra den korrekte fil.

Standarddelen er implementeret som en external procedure - test_print - hvor positioneringen angives i parametrene, mens udvælgelsen af hvilke individer, der skal udskrives, sker ved hjælp af et logisk udtryk, der beregnes for hvert individ (Jensens Device). De udvalgte individer sendes t ad gangen til en udskrivningsprocedure, som er med som parameter til test_print proceduren.

Den brugerafhængige del består af

- program hvorfra test_print proceduren kaldes med passende (= varierende) parametre
- procedure til udskrivning af testindividerne.

2.1 Proceduren test_print

Proceduren test_print positionerer bs-filen med det angivne dokumentnavn til det individ, der er udpeget af rec no, hvorpå individerne læses og overspringes indtil tidsangivelsen i individet er større end eller lig med cur clock. Herefter undersøges udskrivningsbetingelsen og hvis den er sand kaldes proceduren print record; dette gentages for de følgende individer indtil den logiske filafslutning, der indtræffer når blot en af de følgende betingelser er opfyldt:

- fysisk filafslutning
- tidsangivelsen i det aktuelle individ udpeger en dato, der er tidligere end 1. januar 1981 eller senere end dags dato
- tidsangivelsen i det aktuelle individ ligger før det foregående individs tidsangivelse.

Opstår andre statusfejl end end_of_document under læsning af testfilen, afsluttes programmet med standardproceduren std_error.

2.1.1 Kald

Den eksterne procedure test_print kaldes således:

```
test_print(doc, rec_no, coroutine, type, cur_clock,
           print, print_record)
```

hvor

doc

(kald værdi, string)

dokumentnavnet på den bs-fil, der indeholder test individerne. (Læses kun n gang).

rec no

(kald og retur værdi, heltal)

individ nummer

kald værdi:

= 0 start med første individ i filen

> 0 start efter rec_no individer i filen

< 0 start rec_no individer fra den logiske fil afslutning

retur værdi:

individ number på sidst læste individ + 1

coroutine

(retur værdi, heltal)

coroutine identifikation

retur værdi:

coroutine identifikation i det sidst læste individ (6. heltal). NB: Kan være negativ jf. [1]

type

(retur værdi, heltal)

individ type

retur værdi:

individ typen i det sidst læste individ (1. heltal)

cur clock

(kald og retur værdi, heltal)

klokken på formen ttmss

kald værdi:

= 0 start med rec_no individet

> 0 start med det første individ efter rec_no individet,

hvis klokke er større end eller lig med cur_clock

retur værdi:

den (formatterede) klokke i det sidst læste individ (7. + 8. heltal = sidste long)

print

(kald værdi, logisk udtryk)

udskrivningsbetingelse, der udregnes på basis af parametrene `rec_no`, `coroutine`, `type` og `cur_clock`, som opdateres for hvert individ, der læses (Jensens Device)

- sand medfører at proceduren `print_record` kaldes
- falsk medfører at det aktuelle individ springes over

print_record

(procedure)

procedure, der kaldes såfremt udskrivningsbetingelsen for det aktuelle individ er sand. Se desuden afsnit 2.2.

2.1.2 Fejludskrifter

Et kald af proceduren `test_print` kan resultere i følgende udskrifter på current output:

relative record number greater than max record number,
i.e. <kald værdien af `recno`> >= <antal individer>

Denne udskrift kommer, hvis der er <antal individer> i den logiske fil og proceduren er blevet bedt om at starte et større antal individer fra afslutningen. Proceduren fortsætter som om `rec_no` = 0.

file exhausted; record number <antal individer>

Denne udskrift kommer, hvis proceduren er blevet kaldt med en kombination af `rec_no` og `cur_clock`, som ikke kan opfyldes inden den logiske filafslutning. Proceduren afslutter.

2.2 Proceduren `print_record`

Proceduren `print_record` er brugerafhængig og funktionen af den kan derfor ikke beskrives. Det eneste der står fast er, at dens parametre skal erklæres som følger:

```
procedure print_record (record, rec_no, cur_date, cur_clock);
value                rec_no, cur_date, cur_clock ;
integer array        record
integer                rec_no, cur_date, cur_clock ;
```

samt at den kaldes af `test_print`, når udskrivningsbetingelsen (`print`) er sand for det aktuelle individ.

Navnet `print_record` er blot det formelle navn, som er anvendt i erklæringen af proceduren `test_print`; der kan vælges et hvilket som helst navn for udskrivningsproceduren. Parametrene til `print_record` er alle kaldeparametre, som ikke må ændres af proceduren:

record
heltalsvektor (1:8)
det testindivid, der skal udskrives

rec no
heltal
det aktuelle individnummer talt fra starten af den fysiske fil (startende med nummer 1)

cur date
heltal, ååmmdd
den aktuelle dato læst fra record (og fortolket)

cur clock
heltal, ttmss
det aktuelle klokkeslet læst fra record (og fortolket).

Som et eksempel på en brugerdefineret print record er valgt den, der benyttes til testfiler produceret af CENTERNET modulet SMM. I dette tilfælde er proceduren external.

2.2.1 Eksempel på print_record procedure

Den print record procedure, der anvendes til SMM, producerer en udskrift, som er illustreret i eksempel 1.

Eksempel 1: Starten af en udskrivning fra SMM's print_record procedure.

Kommentarer til SMM's print_record procedure:

- I starten af udskriften laves en overskrift, hvoraf de 3 første linier hører til de individer, der dannes af coroutinesystemet, mens de sidste svarer til de brugerafhængige individer.
- Overskriften styres af et antal own variable: last_clock, last_date, tab_1, tab_2 samt width. De samme variable anvendes i de følgende kald til at formattere udskriften.
- Hver gang dato ændrer sig udskrives en linie med den nye dato (cur_date).
- I starten af hver individudskrift er der plads til klokkeslet (cur_clock), som kun udskrives, når det skifter.
- Coroutine identifikationen, individtypen (operation) og semaforidentifikationen udskrives i klar tekst.
- Det aktuelle individnummer udskrives sammen med det ufortolkede individindhold.
- De interessante felter i et individ udskrives i den 'fortolkede' del af udskriften.
- Der anvendes en standard variable terms til oplysning, om det antal terminaler, der blev benyttet ved opstarten af SMM. (Værdien har betydning for fortolkningen af coroutine- og semaforidentifikationerne).
- De to dele af wait select udskrives hver især, enten som ikke-negative tal (dvs. ufortolket) eller som et minus efterfulgt af de sidste 23 bits opfattet som et positivt heltal. Denne metode skulle lette fortolkningen af wait_select i de tilfælde, hvor wait_select opfattes som en maske.

2.3 Program

Hvilket program, der skal udføres for at få kaldt proceduren test_print, er afhængigt af hvordan den aktuelle testfil skal behandles. I det følgende gives et antal eksempler på programmer til illustrering af nogle af de muligheder, der ligger i test_print proceduren.

Det simpleste program er det, hvor samtlige individer fra start og til logisk filafslutning udskrives:

```
begin
    test_print(<:test:>,0,0,0,0,true,print_rec)
end
```

Såfremt udskrivningsproceduren er en med standard variable som for eksempel den, der anvendes til SMM, skal disse variable initieres inden

kaldet af test_print. Hvis der desuden kun ønskes udskrevet fra individ nr. 200 til individ nr. 300 kan det gøres med følgende program:

```
begin
integer rec_no;
  terms:= 3; < * antal terminaler * >
  rec_no:= 199;
  test_print(<:smmtest:>,rec_no,0,0,0,rec_no <= 300,
                                                    print_record);
end
```

Ønskes samtlige 'signal'- og 'wait'-individer udskrevet, såfremt de ikke er genereret af centrallogic (dvs. coroutine nr. 0), kan det ske med følgende program:

```
begin
integer cor,type;
  test_print(<:test:>,0,cor,type,0,
            cor > 0 and (type = 4 or type = 5 or type = 8),
            print_rec);
end
```

Vides det, at tidsintervallet fra klokken 13 til 14 rummes i de sidste 2000 individer i testfilen, og ønskes blot de brugergenererede individer udskrevet fra dette tidsrum, kan det gøres på følgende måde:

```
begin
integer type,cur_clock;
  cur_clock:= 13 00 00;
  test_print(<:fil:>,-2000,0,type,cur_clock,
            type >= 1024 and cur_clock <= 14 00 00,
            print_rec)
end
```

Som det fremgår af ovenstående eksempler, giver Jensens Device rige muligheder for at trimme udskriften af testfiler og samtidig er selekteringsmetoden enkel, idet print parametren blot skal overholde reglerne for logiske udtryk, som kendes fra ALGOL8.

Appendix A. Referencer

Part numbers in references are subject to change as new editions are issued and are listed as an identification aid only. To order, use package number.

- 1) PN: 991 11290
ALGOL Coroutines, User's Guide
Fås med SW8585, Compiler Collection
- 2) PN: 991 07025
CENTERNET, RC8000 SMM

