

TOPPERS/ASPカーネルのトレースログの 標準形式トレースログへの変換ルール例

名古屋大学 大学院情報科学研究科
附属組込みシステム研究センター

はじめに

❖本ドキュメントは、標準形式トレースログへの変換ルールについて、TOPPERS/ASPカーネルのトレースログの変換ルールの一つの例を用いて説明したものである.

目次

- ❖標準形式変換ルールとは？
- ❖ファイル構成
- ❖標準形式変換ルールの作成法
- ❖出力させたいリソースを決める.
- ❖/resourceHeader/asp.reshに, リソースタイプの定義をする.
- ❖/convertRules/asp.cnvに, 標準形式変換ルールを書く.
- ❖置換マクロ
- ❖tlv.tfに各リソースのtemplateを記述をする.
- ❖tlv.tfの文法/実例
- ❖TLVで, 変換された標準形式ログを確認する.

標準形式変換ルールとは？

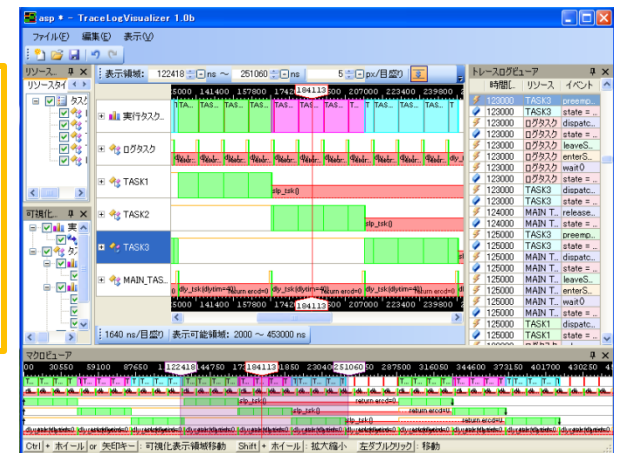
❖ OSが出力した様々のトレースログを，TLVが可視化のために，認識できるような形式にする規則.

```
1 [2000] dispatch to task 1.
2 [2000] enter to ena_int intno=2.
3 [2000] leave to ena_int ercd=0.
4 [2000] enter to dly_tsk dlytim=10.
5 [2000] task 1 becomes WAITING.
6 [2000] dispatch to task 5.
7 [2000] enter to act_tsk tskid=2.
8 [2000] task 2 becomes RUNNABLE.
9 [2000] leave to act_tsk ercd=0.
10 [2000] enter to act_tsk tskid=3.
11 [2000] task 3 becomes RUNNABLE.
12 [2000] leave to act_tsk ercd=0.
13 [2000] enter to act_tsk tskid=4.
14 [2000] task 4 becomes RUNNABLE.
15 [2000] leave to act_tsk ercd=0.
16 [2000] enter to dly_tsk dlytim=40.
```

今回は，標準形式
変換について説明

標準形式変換

```
2 |LOGTASK.dispatch()",
3 |LOGTASK.state=RUNNING",
4 |LOGTASK.enterSVC(ena_int,intno=2)",
5 |LOGTASK.leaveSVC(ena_int,ercd=0)",
6 |LOGTASK.enterSVC(dly_tsk,dlytim=10)",
7 |LOGTASK.wait()",
8 |LOGTASK.state=WAITING",
9 |MAIN_TASK.dispatch()",
10 |MAIN_TASK.state=RUNNING",
11 |MAIN_TASK.enterSVC(act_tsk,tskid=2)",
12 |TASK1.activate()",
13 |TASK1.state=RUNNABLE",
14 |MAIN_TASK.leaveSVC(act_tsk,ercd=0)",
15 |MAIN_TASK.enterSVC(act_tsk,tskid=3)",
16 |TASK2.activate()",
```



可視化変換ルール

ファイル構成

❖TLVの関連ファイル

convertRules > **asp.cnv**, fmp.cnv

resourceHeaders > **asp.resh**, fmp.resh

visualizeRules > asp_rules.viz, asp_shapes.viz,
fmp_rules.viz, fmp_shapes.viz,
toppers_rules.viz, toppers_shapes.viz

❖ASP内の関連ファイル

- ◆ asp/kernel/**kernel.tf**
- ◆ asp/arch/logtrace/**tlv.tf**
- ◆ アプリケーションの**kernel.res**

ASP標準形式変換ルールの作成法

- ❖ 今回はすでに記述されている周期ハンドラの追加方法を例にして説明する.

- ❖ 作成順
 1. 出力させたいリソースを決める.
 2. 周期ハンドラのリソースヘッダを作成.
 3. 周期ハンドラの標準形式変換ルールの作成.
 4. tlv.tfに周期ハンドラのリソースtemplateを記述をする.
 5. TLVで, 変換された標準形式ログを確認する.

1.出力させたいリソースを決める.

asp/arch/logtrace/trace_dump.c

```
2562 /*
2563  * トレースログの表示
2564  */
2565 static void
2566 trace_print(TRACE *p_trace, void (*putc)(char_t))
2567 {
2568     intptr_t    traceinfo[TMAX_LOGINFO + 1];
2569     const char  *tracemsg;
2570     int_t        i;
2571
2572     traceinfo[0] = (intptr_t)(p_trace->logtim);
2573     syslog_printf("[%d]", traceinfo, putc);
2574
2575     switch (p_trace->logtype) {
2576     case LOG_TYPE_INH|LOG_ENTER:
2577         traceinfo[0] = p_trace->loginfo[0];
2578         tracemsg = "enter to int handler %d.";
2579         break;
2580     case LOG_TYPE_INH|LOG_LEAVE:
2581         traceinfo[0] = p_trace->loginfo[0];
2582         tracemsg = "leave from int handler %d.";
2583         break;
2584     case LOG_TYPE_ISR|LOG_ENTER:
2585         traceinfo[0] = p_trace->loginfo[0];
2586         tracemsg = "enter to isr %d.";
2587         break;
2588     case LOG_TYPE_ISR|LOG_LEAVE:
2589         traceinfo[0] = p_trace->loginfo[0];
2590         tracemsg = "leave from isr %d.";
2591         break;
2592     case LOG_TYPE_CYC|LOG_ENTER:
2593         traceinfo[0] = (intptr_t)((ID)((((CYCDB*)(p_trace->loginfo[0])) - cycdb_table) + TMIN_CYCID));
2594         tracemsg = "enter to cyclic handler %d.";
2595         break;
2596     case LOG_TYPE_CYC|LOG_LEAVE:
2597         traceinfo[0] = (intptr_t)((ID)((((CYCDB*)(p_trace->loginfo[0])) - cycdb_table) + TMIN_CYCID));
2598         tracemsg = "leave from cyclic handler %d.";
2599         break;
2600     case LOG_TYPE_ALM|LOG_ENTER:
2601         traceinfo[0] = (intptr_t)((ID)((((ALMDB*)(p_trace->loginfo[0])) - almbc_table) + TMIN_ALMID));
2602         tracemsg = "enter to alarm handler %d.";
2603         break;
2604     case LOG_TYPE_ALM|LOG_LEAVE:
2605         traceinfo[0] = (intptr_t)((ID)((((ALMDB*)(p_trace->loginfo[0])) - almbc_table) + TMIN_ALMID));
2606         tracemsg = "leave from alarm handler %d.";
2607         break;
```

アプリケーションの
xxx.logファイル

```
[41335996] #1#raise CPU exception
[41336332] enter to exc handler 2.
[41337044] CPU exception handler (p_excinfo = 001019d8).
[41337355] enter to sns_tex.
[41337498] leave from sns_tex state=0.
[41338261] sns_loc = 0 sns_dsp = 0 sns_tex = 0
[41338514] enter to xsns_dpn p_excinfo=1055192.
[41338674] leave from xsns_dpn state=0.
[41338819] enter to xsns_xpn p_excinfo=1055192.
[41338977] leave from xsns_xpn state=0.
[41339563] xsns_dpn = 0 xsns_xpn = 0
[41339816] enter to xsns_xpn p_excinfo=1055192.
[41339974] leave from xsns_xpn state=0.
[41340172] enter to iras_tex tskid=2. rasptn=0.
[41340386] leave from iras_tex ercd=0.
[41340537] leave from exc handler 2.
[41340826] enter to tex 32768.
[41341499] task1 receives exception 0x8000.
[41457603] enter to int handler 5.
[41457827] leave from int handler 5.
[41684061] enter to int handler 5.
[41684485] enter to cyclic handler 1.
[41684909] Run cyclic_handler!
[41685163] enter to irot_rdq tskpri=10.
[41685400] leave from irot_rdq ercd=0.
[41685551] leave from cyclic handler 1.
[41685765] leave from int handler 5.
[41686019] dispatch from task 2.
[41686166] dispatch to task 3.
[41686337] leave from slp_tsk ercd=0.
[41686992] task2 is running (002).
[41910271] enter to int handler 5.
[41910495] leave from int handler 5.
[42136729] enter to int handler 5.
[42136953] leave from int handler 5.
[42363189] enter to int handler 5.
[42363413] leave from int handler 5.
[42589649] enter to int handler 5.
[42589873] leave from int handler 5.
[42816107] enter to int handler 5.
[42816331] leave from int handler 5.
[43042566] enter to int handler 5.
[43042790] leave from int handler 5.
```

2. 周期ハンドラのリソースヘッダを作成する.

❖ /resourceHeader/asp.reshtにリソースタイプの定義をする.

❖ リソースタイプの定義

```
“リソースタイプ名” : {  
    “DisplayName” : 表示名  
    “Attributes” : 属性  
    “Behaviors” : 振舞い  
}
```

```
❖ “CyclicHandler” : {  
    “DisplayName” : “周期ハンドラ”,  
    “Attributes” : {  
        “state” : {  
            “VariableType” : “String”,  
            “DisplayName” : “状態”,  
            “AllocationType” : “Dynamic”,  
            “CanGrouping” : false  
        },  
        “id” : {  
            “VariableType” : “Number”,  
            “DisplayName” : “ID”,  
            “AllocationType” : “Static”,  
            “CanGrouping” : false  
        }  
    }  
    “Behaviors” : { /*省略*/ }  
}
```

3. 周期ハンドラの標準形式変換ルールの作成

❖ /convertRules/asp.cnvに、標準形式変換ルールを書く。

❖ [time] enter to cyclic handler x.



正規表現式に直す。

❖ “¥[(?<time>¥d+)¥] enter to cyclic handler (?<id>[^\\$.]+)”



トレースログがこの正規表現に一致する時

❖ “ [time]\$RES_NAME{CyclicHandler(id==id)}.enter()”を
標準形式ログとして出力する。

❖ “[time]CurrentContext.name=\$RES_NAME{CyclicHandler(id==id)}”の解説は省略する。

3.周期ハンドラの標準形式変換ルールの作成

❖置換マクロ

❖\$EXIST{リソース}  該当するリソースが存在したら、`true`を、存在しなかったら`false`をリターン.

❖\$ATTR{属性}  該当する属性に置き換えられる.

❖\$RES_COLOR{リソース}  xxx.resで定義している、各リソースの色に置き換えられる.

❖\$RES_NAME{リソース}  リソース名に置き換えられる.

4.tlv.tfに周期ハンドラのリソースtemplateを記述

- ❖ リソースファイルを生成するTFファイルを作成.
- ❖ 今回は ‘CyclicHandler’ リソースの追加法を説明しながら, tlv.tfの文法を簡単に説明する.

4.tlv.tfに周期ハンドラのリソースtemplateを記述

asp/arch/logtrace/tlv.tf

```
❖$NL$ ← 改行の出力
❖$JOINEACH cycid CYC.ID_LIST
  ",¥n"$ ← TABの出力
❖$TAB$$TAB$"$cycid$":{$NL$
❖$TAB$$TAB$$TAB$"Type"
  : "CyclicHandler", $NL$
❖$TAB$$TAB$$TAB$"Attribute..."
  : $NL$
❖$TAB$$TAB$$TAB${$NL$
❖$TAB$$TAB$$TAB$$TAB$"id"
  : $+cycid$, $NL$ ← VALUEの値
❖$TAB$$TAB$$TAB$$TAB$"state"
  : "Dormant" $NL$ ← stateの初期状態
❖$TAB$$TAB$$TAB$} $NL$
❖$TAB$$TAB$}
❖$END$
```

\$で始まり\$で終わる:マクロ命令
\$空白 で始まる :コメント
何もなし :そのまま出力

繰り返し構文

\$JOINEACH <変数> <順序付きリスト> "区切り文字" \$
<繰り返し記述>
\$END\$

すべてのオブジェクトにあるパラメータ
オブジェクト名.ID_LIST
:割り当てられたID

4.tlv.tfに周期ハンドラのリソースtemplateを記述.

The image shows a development environment with two panes. The left pane displays the source code in `tlv.tf`, and the right pane displays the generated code in `kernel.res`.

Source Code (tlv.tf):

```
527 NL$  
528  
529  
530 ノンドラID番号の最大値,  
531 _kernel_tmax_cycid = (TMIN_CYCID + TNUM_CYCID - 1);$NL$  
532  
533  
534 ノンドラ初期化テーブルの生成,  
535 TH(CYC.ID_LIST)$  
536 const CYCINIB _kernel_cycinib_table[TNUM_CYCID] = {$NL$  
537 $JOINEACH cycid CYC.ID_LIST ",wn"$  
538 // cycatが ( [TA_STA] ) でない場合 (E_RSATR)  
539 $IF (CYC.CYCATR[cycid] & ~TA_STA) != 0$  
540 $ERROR CYC.TEXT_LINE[cycid]$E_RSATR: $FORMAT(  
541 $END$  
542  
543 // (0 < cycetim && cycetim <= TMAX_RELTIM)でない場合 (E_  
544 $IF !(0 < CYC.CYCTIM[cycid] && CYC.CYCTIM[cycid] <= Tm  
545 $ERROR CYC.TEXT_LINE[cycid]$E_PAR: $FORMAT(  
546 $END$  
547  
548 // (0 <= cycphs && cycphs <= TMAX_RELTIM)でない場合 (E_  
549 $IF !(0 <= CYC.CYCPHS[cycid] && CYC.CYCPHS[cycid] <= T  
550 $ERROR CYC.TEXT_LINE[cycid]$E_PAR: $FORMAT(  
551 $END$  
552  
553 // 警告: cycatにTA_STAが設定されていて, (cycphs == 0)  
554 $IF (CYC.CYCATR[cycid] & TA_STA) != 0 && CYC.CYCPHS[cy  
555 $WARNING CYC.TEXT_LINE[cycid]$$FORMAT(("%1% i  
556 $END$  
557  
558 // 周期ハンドラ初期化ブロック  
559 $TAB$(CYC.CYCATR[cycid]$), (intptr_t)($CYC.EXINF[cy  
560 $END$$NL$  
561 }:$NL$  
562 $NL$  
563  
564 // 周期ハンドラコントロールブロック  
565 CYCCB _kernel_cyccb_table[TNUM_CYCID]:$NL$  
566  
567 TOPPERS_EMPTY_LABEL(const CYCINIB, _kernel_cycinib_table);$NL$  
568 TOPPERS_EMPTY_LABEL(CYCCB, _kernel_cyccb_table);$NL$  
569  
570
```

Generated Code (kernel.res):

```
58 $TAB$$TAB$$TAB$$TAB$state" : "DORMANT"$NL$  
59 $TAB$$TAB$$TAB$}$NL$  
60 $TAB$$TAB$}  
61 $END$  
62 $NL$  
63 $JOINEACH cycid CYC.ID_LIST ",wn"$  
64 $TAB$$TAB$$CYCID$":{$NL$  
65 $TAB$$TAB$$TAB$Type": "CyclicHandler", $NL$  
66 $TAB$$TAB$$TAB$Attributes": $NL$  
67 $TAB$$TAB$$TAB$}$NL$  
68 $TAB$$TAB$$TAB$$id" : 1, $NL$  
69 $TAB$$TAB$$TAB$$state" : "DORMANT"$NL$  
70 $TAB$$TAB$$TAB$}$NL$  
71 $TAB$$TAB$}  
72 $END$  
73 $NL$  
74 $JOINEACH almid ALM.ID_LIST ",wn"$  
75 $TAB$$TAB$$Almid$":{$NL$  
76 $TAB$$TAB$$TAB$Type": "AlarmHandler", $NL$  
77 $TAB$$TAB$$TAB$Attributes": $NL$  
78 $TAB$$TAB$$TAB$}$NL$  
79 $TAB$$TAB$}  
80 $END$  
81 $NL$  
82 $TAB$$TAB$}  
83 $END$  
84 $NL$  
85 $TAB$$TAB$}  
86 $END$  
87 $NL$  
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 }
```

Text Box:

コンパイルすると,
kernel.resが生成される.

5.TLVで変換された標準形式ログを確認する

トレースログビューア			時間[us]	リソース	イベント
			38649838	MAIN TASK	state = RUN...
			38650003	MAIN TASK	leaveSVC(wa...
			38650373	MAIN TASK	enterSVC(wa...
			38650767	MAIN TASK	wait0
			38650767	MAIN TASK	state = WAIT...
			38651076	TASK1	dispatch0
			38651076	TASK1	state = RUN...
			41006575	TASK1	preempt0
			41006575	TASK1	state = RUN...
			41007677	TASK1	dispatch0
			41007677	TASK1	state = RUN...
			41336332	CPUEXC1	enter0
			41336332	CPUEXC1	state = RUN...
			41337355	CPUEXC1	enterSVC(sn...
			41337498	CPUEXC1	leaveSVC(sn...
			41338514	CPUEXC1	enterSVC(xs...
			41338674	CPUEXC1	leaveSVC(xs...
			41338819	CPUEXC1	enterSVC(xs...
			41338977	CPUEXC1	leaveSVC(xs...
			41339816	CPUEXC1	enterSVC(xs...
			41339974	CPUEXC1	leaveSVC(xs...
			41340172	CPUEXC1	enterSVC(ira...
			41340386	CPUEXC1	leaveSVC(ira...
			41340537	CPUEXC1	leave0
			41340537	CPUEXC1	state = DOR...
			41340826	TASK_TEX	enter0
			41340826	TASK_TEX	state = RUN...
			41684485	CYCHDR1	enter0
			41684485	CYCHDR1	state = RUN...
			41685163	CYCHDR1	enterSVC(iro...
			41685163	CYCHDR1	leaveSVC(iro...
			41685551	CYCHDR1	leave0
			41685551	CYCHDR1	state = DOR...
			41686166	TASK1	preempt0
			41686166	TASK1	state = RUN...
			41686166	TASK2	dispatch0
			41686166	TASK2	state = RUN...
			41686337	TASK2	leaveSVC(slp...
			43496504	TASK2	preempt0
			43496504	TASK2	state = RUN...
			43700864	TASK2	dispatch0
			43700864	TASK2	state = RUN...
			44849730	ALMHDR1	enter0
			44849730	ALMHDR1	state = RUN...
			44850395	ALMHDR1	enterSVC(iro...
			44850632	ALMHDR1	leaveSVC(iro...
			44850783	ALMHDR1	leave0
			44850783	ALMHDR1	state = DOR...
			44851398	TASK2	preempt0
			44851398	TASK2	state = RUN...
			44851398	TASK3	dispatch0
			44851398	TASK3	state = RUN...
			44851569	TASK3	leaveSVC(slp...
			45982420	TASK3	preempt0
			45982420	TASK3	state = RUN...
			46032098	TASK3	dispatch0
			46032098	TASK3	state = RUN...

❖変換に成功すると，交通形式ログがファイル化され，このように出力される．