

信学技報 FIIS-24, No.600

UNIX第5版における プロセススイッチングの可視化

いわしまふうや しらいちさと たむらおさむ くどうしんいちろう
岩島楓也, 白井千智, 田村修, 工藤信一朗 (金沢工大)
いしいみつる たかごうだいすけ
石井 充 (関東学院大), 鷹合大輔 (金沢工大)

令和6年6月28日(金)

1. はじめに

- UNIX系OSは企業や大学などで多く活用されている.
- UNIX第1版(1971)から第4版(1973)関連のソースの多くが失われた.
- **初期のUNIXで使われた技術の保存**を目的として, ①カーネルソースの解析と, ②解析支援プログラム（トレーサ）の開発を進めている.
 - 「UNIX第1版におけるタスク切り替え機構の解析」(FIIS-22-558) [1]
 - 「初期UNIXにおけるプロセス切り替え機構の動的解析」(FIIS-23-570) [2]

■ 今回は**第5版用の新トレーサ**について発表

- 第5版 (1974, [3]) はカーネルソースとファイルシステムが現存. シミュレータ上で実行可能.
- 旧トレーサと比べ, プロセス情報をより詳細に表示可能.

```
login: root
#
# cat > test.c
main(){
    printf("Hello World!!\n");
}
# cc test.c
# ls -l a.out
-rwxrwxrwx 1 root      1204 Mar 21 13:48 a.out
# ./a.out
Hello World!!
#
#
```

UNIX第5版(1973)
DEC PDP-11/45, 11/4
メインメモリ 144KB
実アドレス空間 256KB

発表の流れ

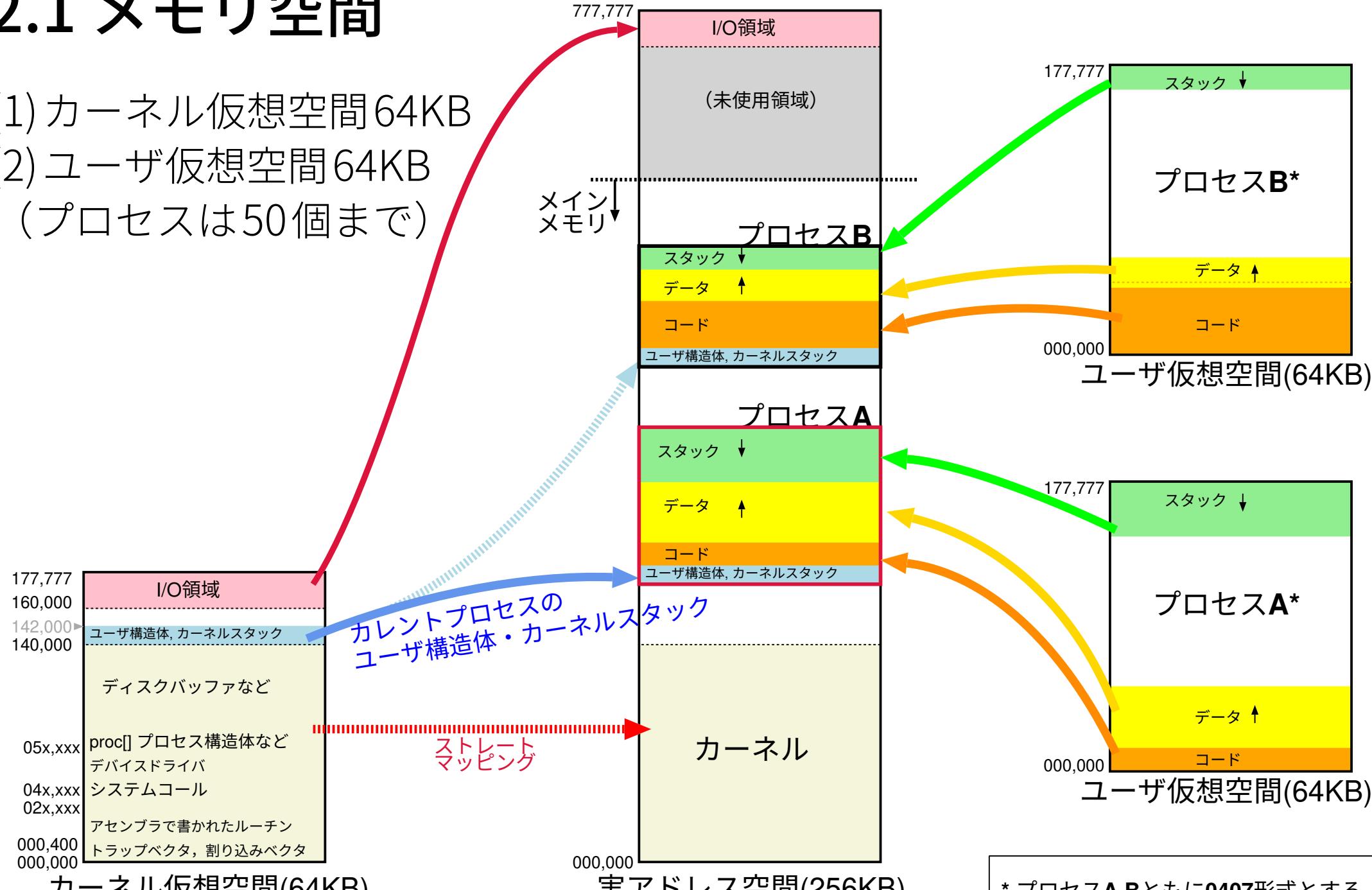
1. はじめに
2. UNIX第5版におけるプロセス管理
3. 新トレーサの紹介
4. 新トレーサによるトレース事例
5. むすび

2. UNIX第5版におけるプロセス管理

1. メモリ空間
2. プロセス構造体とユーザ構造体
3. スワップ領域
4. プロセスの生成と終了に関するシステムコール

2.1 メモリ空間

- (1) カーネル仮想空間 64KB
- (2) ユーザ仮想空間 64KB
(プロセスは50個まで)



* プロセスA,Bともに0407形式とする

2.2 ユーザ構造体とプロセス構造体

ユーザ構造体 (sys/user.h の struct user) カレントプロセスに関連する情報（プロセス構造体のアドレス, データサイズ, スタックサイズなど）を格納する。カーネル仮想空間上の固定位置にマッピングされる。

プロセス構造体 (sys/proc.h の struct proc) プロセス状態, フラグ, PID, プロセスに割り当てられたメモリアドレスとサイズ等を格納する。カーネル仮想空間上に50個分用意される。

これらの構造体にはコマンド名に関する情報は含まれない（コマンドはスタック領域の最下部に置かれている）。

2.3 スワップ領域

1. メモリの空き領域の確保
 - メモリの空き領域が足りず、プロセスを配置できなくなったときに、CPUを割り当てていないプロセスをスワップアウト
2. プロセスのスタック領域を拡張する際の作業領域
 - スタック領域を拡張するプロセスをスワップアウトし、スワップ領域で拡張したあとにメモリに再配置する
3. プロセスの終了処理の一部

2.4 プロセスの生成と終了に関するシステムコール

■ 生成

fork 親プロセスのプロセス構造体とデータセグメントをコピーする。ただし、PIDは新しいものに差し替えられ、PPIDには親プロセスのPIDが格納される。

exec プロセスイメージを実行可能ファイルに差し替える。コマンドの情報はプロセス構造体やユーザ構造体には格納されない。

■ 終了

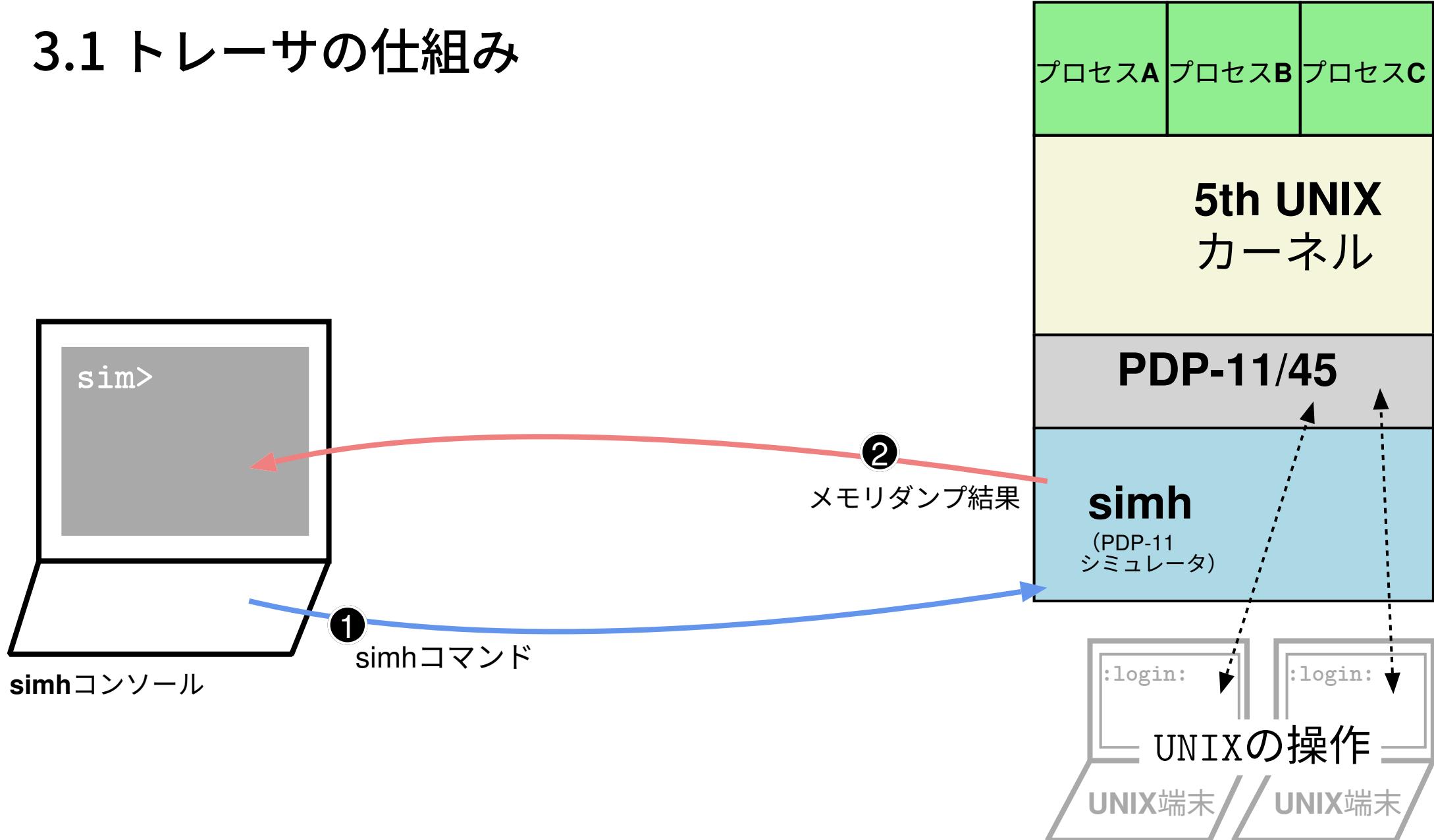
exit プロセス状態をゾンビにセットし、initと親プロセスをウェイクアップにセットする。子プロセスがある場合は、initを親プロセスにセットする。

wait ゾンビプロセスの開放を行い、スリープ状態で子プロセスの終了を待つ。ゾンビプロセスの開放処理では、プロセス構造体の初期化、スワップの開放が行われる。

3. 新トレーサの紹介

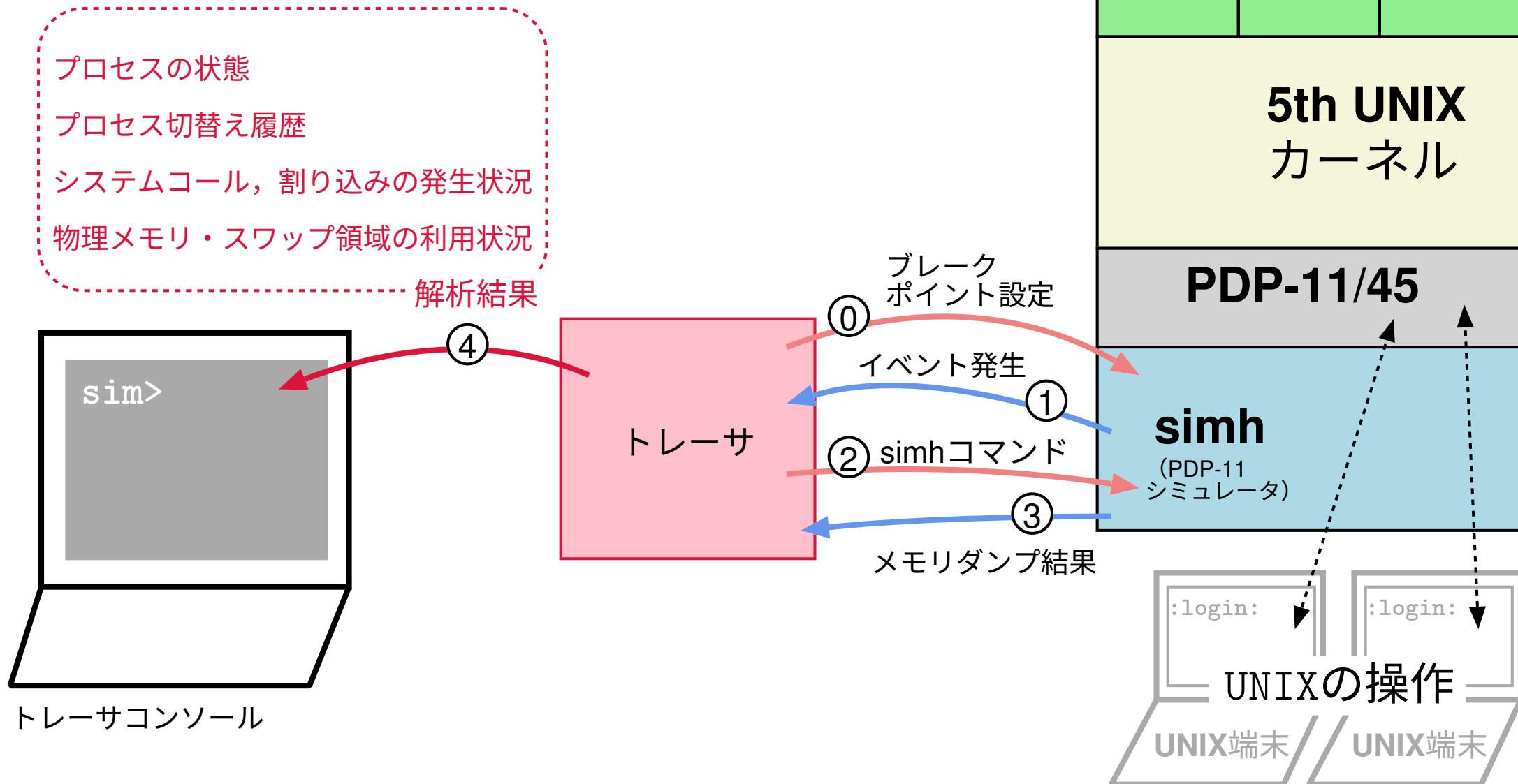
1. トレーサの仕組み
2. 新トレーサと旧トレーサの比較
3. 新トレーサの各画面の見方

3.1 トレーサの仕組み



メモリをダンプしても、ダンプ結果がUNIXの何に関する情報かは特定できない（特定には多くの作業が必要）。

3.1 トレーサの仕組み



トレーサはsimh[4]を自動的に制御して、UNIXの内部情報の抽出、可視化を行う。見かけ上はUNIXを稼働させたままでトレース。

3.2 旧トレーサ

トレーサコンソール

```

1374 ##### CONTEXT_SW ##### PID:0
1374 ##### CONTEXT_SW ##### PID:19

1374 ##### TRAP 0 (=nullsys) ##### PID:19
1374 IND_SYSCALL 10 (=unlink) PID:19      arg0 : test.o

1374 ##### TRAP 0 (=nullsys) ##### PID:19
1374 IND_SYSCALL 9 (=link) PID:19        arg0 : a.out
                                         arg0 : a.out

1374 ##### CONTEXT_SW ##### PID:0
1374 ##### RK11 #####
1374 ##### CONTEXT_SW ##### PID:0
1374 ##### CONTEXT_SW ##### PID:19

1374 ##### TRAP 0 (=nullsys) ##### PID:19
1374 IND_SYSCALL 10 (=unlink) PID:19      arg0 : a.out

1374 ##### KW11L #####

```

システムコール
I/O完了割り込み
プロセス切り替え
タイム割り込み

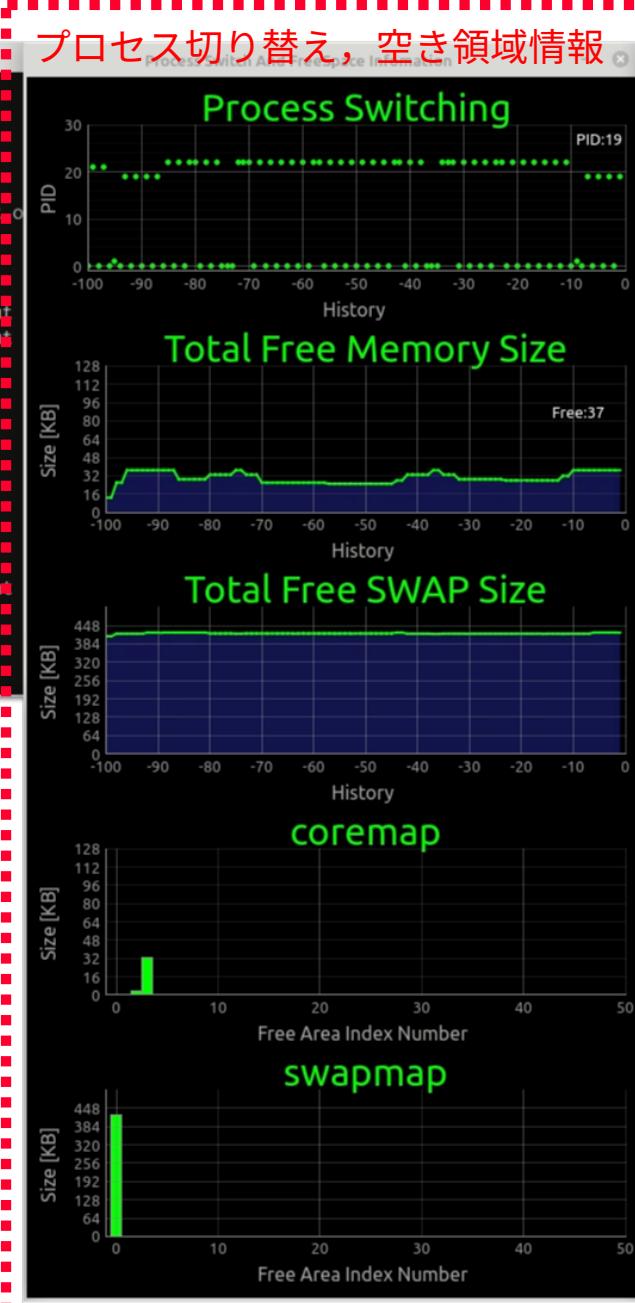
UNIXコンソール

```

./boot.sh
# cat test.c
int pid;

main(){
    int x[8192]; /* 16KB */
    int y[8192]; /* 16KB */
    int z[8192]; /* 16KB */
    pid=getpid();
    printf("pid:%d %o(%8), addr:%o(%8)\n", pid,pid,&pid);
    while(1);
}
# cc test.c

```



3.2 新トレーサ

	構造体番号	コマンド	コンテキストスイッチ数	プロセス状態	メモリアドレス	サイズ		
	PID	PPID	cmd	cx	pri	stat	flag	mem(oct)
0	0	0	[sched]	17	-100	slep	LS--	0116100-0120077(1.0KB)
1	1	0	/etc/init	15	40	wait	L---	0125500-0132677(2.6KB)
2	7	1	/etc/init	4	10	wait	L---	0132700-0140077(2.6KB)
3	8	1	/etc/init	2	10	wait	L---	0140100-0145277(2.6KB)
4	6	1	/etc/update	3	90	wait	L---	0165300-0171777(2.3KB)
5	9	1	/etc/init	2	10	wait	L---	A145200-A152477(2.6KB)
6	10	1	/etc/i					6KB)
7	11	1	/etc/i					6KB)
8	12	1	/etc/i...					6KB)
9	13	1	/etc/init	2	10	wait	L---	0177200-0204377(2.6KB)
10	14	1	/etc/init	2	10	wait	L---	0204400-0211577(2.6KB)
11	15	1	-	58	100	run	L---	0233500-0244177(4.3KB)
12	0	0	none	0	-50	free	---	0000000-0000000(0.0KB)
13	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
14	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
15	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
16	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
17	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
18	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
19	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
20	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
21	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
22	0	0	none	0	0	free	----	0000000-0000000(0.0KB)
23	0	0	none	0	0	free	----	0000000-0000000(0.0KB)

①プロセス構造体リスト

カレントプロセス

緑：オンメモリ
青：スワップアウト
赤：ゾンビ
灰：空き

トレーサコンソール								
login: root								UNIXコンソール
# ps								No swap
f: 0 ??Zf????hb??bcq??Z? ??P??								q? ?
f: 1 /etc/init								f: 7 /etc/init
f: 8 /etc/init								f: 6 /etc/update
f: 9 /etc/init								f: 10 /etc/init
f: 11 /etc/init								f: 12 /etc/init
f: 13 /etc/init								f: 14 /etc/init
f: 15 -								16 ps
#								#

[proc] FREE RAM 51.1KB, FREE SWAP 429.0KB(858blk), CX 243

	pid	ppid	cmd	cx	pri	stat	flag	mem(oct)
12	16	15	ps	59	101	run	L---	0244200-0254177(4.0KB) 30.152991
12	16	15	ps	60	20	run	L---	0244200-0254177(4.0KB) 30.281457
12	16	15	ps	61	100	run	L---	0244200-0254177(4.0KB) 30.287294
12	16	15	ps	62	101	run	L---	0244200-0254177(4.0KB) 30.404318
12	16	15	ps	63	20	run	L---	0244200-0254177(4.0KB) 30.531691
12	16	15	ps					0KB) 30.537802
12	16	15	ps					0KB) 30.657661
12	16	15	ps	66	20	run	L---	0244200-0254177(4.0KB) 30.787591
12	16	15	ps	67	100	run	L---	0244200-0254177(4.0KB) 30.793536
12	16	15	ps	68	101	run	L---	0244200-0254177(4.0KB) 30.912070
12	16	15	ps	69	20	run	L---	0244200-0254177(4.0KB) 31.044848
12	16	15	ps	70	100	run	L---	0244200-0254177(4.0KB) 31.051073
12	16	15	ps	71	101	run	L---	0244200-0254177(4.0KB) 31.176800
12	16	15	ps	72	101	run	L---	0244200-0254177(4.0KB) 31.313525
12	16	15	ps	73	101	run	L---	0244200-0254177(4.0KB) 31.409009
12	16	15	ps	74	101	run	L---	0244200-0254177(4.0KB) 31.505974
12	16	15	ps	75	-50	run	L---	0244200-0254177(4.0KB) 31.525039
1	1	0	/etc/init	15	40	run	L---	0125500-0132677(2.6KB) 31.532073
11	15	1	-	57	40	run	L---	0233500-0244177(4.3KB) 31.538011
11	15	1	-	58	100	run	L---	0233500-0244177(4.3KB) 31.544727

②コンテキストスイッチ履歴

[hist] FREE RAM 51.1KB, FREE SWAP 429.0KB(858blk), CX 243

表示内容 空きメモリ 空きスワップ コンテキストスイッチ数(累計) タイムスタンプ

	pid	ppid	cmd	cx	pri	stat	flag	mem(oct)
0	0	0	[kernel]	##KERNEL##				# 0000000-0116077(39.1KB)
				FREE MEMORY				* 0123500-0125477(1.0KB)
1	1	0	/etc/init	15	40	wait	L---	0125500-0132677(2.6KB)
2	7	1	/etc/init	4	10	wait	L---	0132700-0140077(2.6KB)
3	8	1	/etc/init					45277(2.6KB)
5	9	1	/etc/init					52477(2.6KB)
6	10	1	/etc/init	2	10	wait	L---	0152500-0157677(2.6KB)
7	11	1	/etc/init	2	10	wait	L---	0157700-0165077(2.6KB)
				FREE MEMORY				* 0165100-0165277(0.1KB)
4	6	1	/etc/update	3	90	wait	L---	0165300-0171777(2.3KB)
8	12	1	/etc/init	2	10	wait	L---	0172000-0177177(2.6KB)
9	13	1	/etc/init	2	10	wait	L---	0177200-0204377(2.6KB)
10	14	1	/etc/init	2	10	wait	L---	0204400-0211577(2.6KB)
				FREE MEMORY				* 0223400-0233477(4.1KB)
11	15	1	-	58	100	run	L---	0233500-0244177(4.3KB)
				FREE MEMORY				* 0244200-0377777(45.9KB)
				##RESERVED##				# 0400000-0757777(120.0KB)
				##I/O##				# 0760000-0777777(8.0KB)

[memm] FREE RAM 51.1KB, FREE SWAP 429.0KB(858blk), CX 243

アドレス サイズ

アドレス サイズ

3.3 新トレーサの各画面の見方

構造体番号
PID PPID
コマンド
コンテクストスイッチ数
プライオリティ
プロセス状態
フラグ
メインメモリアドレス
又はスワップ先
サイズ

pid	ppid	cmd	cx	pri	stat	flag	mem(oct)	
0	0	[sched]	17	-100	slep	LS--	0116100-0120077(1.0KB)
1	1	/etc/init	15	40	wait	L---	0125500-0132677(2.6KB)
2	7	/etc/init	4	10	wait	L---	0132700-0140077(2.6KB)
3	8	/etc/init	2	10	wait	L---	0140100-0145277(2.6KB)
4	6	/etc/update	3	90	wait	L---	0165300-0171777(2.3KB)
5	9	/etc/init	?	10	wait	L---	0145300-0152477(2.6KB)
6	10	/etc/i						6KB)
7	11	/etc/i						6KB)
8	12	/etc/i...	?	10	wall	L---	0112000-0111111(2.6KB)
9	13	/etc/init	2	10	wait	L---	0177200-0204377(2.6KB)
10	14	/etc/init	2	10	wait	L---	0204400-0211577(2.6KB)
11	15	1 -	58	100	run	L---	0233500-0244177(4.3KB)
12	0	none	0	-50	free	----	0000000-0000000(0.0KB)
13	0	none	0	0	free	----	0000000-0000000(0.0KB)
14	0	none	0	0	free	----	0000000-0000000(0.0KB)
15	0	none	0	0	free	----	0000000-0000000(0.0KB)
16	0	none	0	0	free	----	0000000-0000000(0.0KB)
17	0	none	0	0	free	----	0000000-0000000(0.0KB)
18	0	none	0	0	free	----	0000000-0000000(0.0KB)
19	0	none	0	0	free	----	0000000-0000000(0.0KB)
20	0	none	0	0	free	----	0000000-0000000(0.0KB)
21	0	none	0	0	free	----	0000000-0000000(0.0KB)
22	0	none	0	0	free	----	0000000-0000000(0.0KB)
23	0	none	0	0	free	----	0000000-0000000(0.0KB)
[proc] FREE RAM 51.1KB, FREE SWAP 429.0KB(858blks), CX 243								

① プロセス構造体リスト

緑：オンメモリ
青：スワップアウト
赤：ゾンビ
灰：空き

カレントプロセス

3.3 新トレーサの各画面の見方

pid	ppid	cmd	cx	pri	stat	flag	mem(oct)	
12	16	15 ps	59	101	run	L---	0244200-0254177(4.0KB)	30.152991
12	16	15 ps	60	20	run	L---	0244200-0254177(4.0KB)	30.281457
12	16	15 ps	61	100	run	L---	0244200-0254177(4.0KB)	30.287294
12	16	15 ps	62	101	run	L---	0244200-0254177(4.0KB)	30.404318
12	16	15 ps	63	20	run	L---	0244200-0254177(4.0KB)	30.531691
12	16	15 ps					0KB)	30.537802
12	16	15 ps					0KB)	30.657661
12	16	15 ps	66	20	run	L---	0244200-0254177(4.0KB)	30.787591
12	16	15 ps	67	100	run	L---	0244200-0254177(4.0KB)	30.793536
12	16	15 ps	68	101	run	L---	0244200-0254177(4.0KB)	30.912070
12	16	15 ps	69	20	run	L---	0244200-0254177(4.0KB)	31.044848
12	16	15 ps	70	100	run	L---	0244200-0254177(4.0KB)	31.051073
12	16	15 ps	71	101	run	L---	0244200-0254177(4.0KB)	31.176800
12	16	15 ps	72	101	run	L---	0244200-0254177(4.0KB)	31.313525
12	16	15 ps	73	101	run	L---	0244200-0254177(4.0KB)	31.409009
12	16	15 ps	74	101	run	L---	0244200-0254177(4.0KB)	31.505974
12	16	15 ps	75	-50	run	L---	0244200-0254177(4.0KB)	31.525039
1	1	/etc/init	15	40	run	L---	0125500-0132677(2.6KB)	31.532073
11	15	1 -	57	40	run	L---	0233500-0244177(4.3KB)	31.538011
11	15	1 -	58	100	run	L---	0233500-0244177(4.3KB)	31.544727
[hist] FREE RAM 51.1KB, FREE SWAP 429.0KB(858blks), CX 243								

表示内容 空きメモリ 空きswap コンテキストスイッチ数(累計) タイムスタンプ

3.3 新トレーサの各画面の見方

pid	ppid	cmd	cx	pri	stat	flag	mem(oct)
		##KERNEL##					カーネル
0	0	[sched]	17	-100	slep	LS--	# 0000000-0116077(39.1KB)
		FREE MEMORY					システムプロセス 空きメモリ
1	1	/etc/init	15	40	wait	L---	* 0123500-0125477(1.0KB)
2	7	/etc/init	4	10	wait	L---	* 0132700-0140077(2.6KB)
3	8	/etc/init	2	10	wait	L---	45277(2.6KB)
5	9	/etc/init	2	10	wait	L---	52477(2.6KB)
6	10	/etc/init	2	10	wait	L---	0152500-0157677(2.6KB)
7	11	/etc/init	2	10	wait	L---	0157700-0165077(2.6KB)
		FREE MEMORY					* 0165100-0165277(0.1KB)
4	6	/etc/update	3	90	wait	L---	0165300-0171777(2.3KB)
8	12	/etc/init	2	10	wait	L---	0172000-0177177(2.6KB)
9	13	/etc/init	2	10	wait	L---	0177200-0204377(2.6KB)
10	14	/etc/init	2	10	wait	L---	0204400-0211577(2.6KB)
		FREE MEMORY					* 0223400-0233477(4.1KB)
11	15	-	58	100	run	L---	0233500-0244177(4.3KB)
		FREE MEMORY					* 0244200-0377777(45.9KB)
		##RESERVED##					# 0400000-0757777(120.0KB)
		##I/O##					# 0760000-0777777(8.0KB)
[memm] FREE RAM 51.1KB, FREE SWAP 429.0KB(858blks), CX 243							

アドレス サイズ

4. 新トレーサによるトレース事例

1. ログインとシェルの動作
2. タイマ割込みに伴うプロセススイッチング
3. メモリ不足時のプロセス再配置
4. ゾンビプロセスの発生，親プロセスによるリソース開放



【事例①】ログインとシェルの動作

pid	ppid	cmd	cx	pri	stat	flag	mem(oct)	
11	15	1 /bin/login root	35	-50	run L---	0217700-0230477(4.4KB)	174.251608
11	15	1 /bin/login root	36	100	run L---	0217700-0230477(4.4KB)	174.255717
11	15	1 /bin/login root	37	-50	run L---	0217700-0230477(4.4KB)	174.295600
11	15	1 /bin/login root	38	-50	run L---	0217700-0230477(4.4KB)	174.302542
11	15	1 /bin/login root	39	100	run L---	0217700-0230477(4.4KB)	174.306960
11	15	1 /bin/login root	40	-50	run L---	0217700-0230477(4.4KB)	174.339673
11	15	1 /bin/login root	41	-50	run L---	0217700-0230477(4.4KB)	174.345173
11	15	1 /bin/login root	42	-50	run L---	0217700-0230477(4.4KB)	174.353308
11	15	1 /bin/login root	43	-50	run L---	0217700-0230477(4.4KB)	174.360727
11	15	1 /bin/login root	44	-50	run L---	0217700-0230477(4.4KB)	174.369139
11	15	1 /bin/login root	45	-50	run L---	0217700-0230477(4.4KB)	174.377276
11	15	1 /bin/login root	46	-100	run L-l-	0221700-0235477(5.9KB)	174.412057
0	0	[sched]	17	-100	run LS--	0116100-0120077(1.0KB)	174.421064
11	15	1 /bin/login root	47	-100	run L---	0223400-0225377(1.0KB)	174.430562
11	15	1 /bin/login root	48	-50	run L---	0225400-0233477(3.1KB)	174.447696
11	15	1 -	49	100	run L-	ログインシェル(PID:15)の実行が始まった		
11	15	1 -	50	-50	run L-			
11	15	1 -	51	-50	run L---	0225400-0233477(3.1KB)	174.478940
11	15	1 -	52	-50	run L---	0225400-0233477(3.1KB)	174.487552
11	15	1 -	53	-50	run L---	0225400-0233477(3.1KB)	174.498508
11	15	1 -	54	100	run L---	0225400-0233477(3.1KB)	174.504598
11	15	1 -	55	10	r-----			
11	15	1 -	56	100	r-----	ログインシェルの子プロセス(PID:16)が生成された		
12	16	15 -	1	0	run L---	0244200-0254677(4.3KB)	183.881918
12	16	15 -	2	101	run L---	0244200-0254677(4.3KB)	183.896525
12	16	15 -	3	-50	run L---	0244200-0254677(4.3KB)	183.907016
12	16	15 -	4	-50	run L---	0244200-0254677(4.3KB)	183.913587
12	16	15 -	5	-50	run L---	0244200-0254677(4.3KB)	183.923007
12	16	15 -	6	-100	run L-l-	0223400-0227077(1.8KB)	183.939234
0	0	[sched]	18	-100	run LS--	0116100-0120077(1.0KB)	183.950680
12	16	15 -	7	-100	run L---	0225100-0227077(1.0KB)	183.960086
12	16	15 echo Hello	8	100	run L---	プロセス(PID:16)のプロセスイメージがechoに置き換わった		
12	16	15 echo Hello	9	-50	run L---			
12	16	15 echo Hello	10	-50	run L---	0244200-0251777(2.9KB)	184.008674
1	1	0 /etc/init	15	40	run L---	プロセス(PID:16)が終了してinitと親プロセス(PID:15)が呼び出された		
11	15	1 -	57	40	run L---	0233500-0244177(4.3KB)	184.021732
11	15	1 -	58	100	run L---			
[hist] FREE RAM 51.1KB, FREE SWAP 429.0KB(858blks), CX 195								

UNIXコソール

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connected to the PDP-11 simulator CON-TELNET device
@unix

login: root ← ログイン
echo Hello ← echoコマンド実行
Hello
█

ログインシェル(PID:15)の実行が始まった

ログインシェルの子プロセス(PID:16)が生成された

プロセス(PID:16)のプロセスイメージがechoに置き換わった

プロセス(PID:16)が終了してinitと親プロセス(PID:15)が呼び出された

【事例②】タイマ割込みに伴うプロセススイッチング



スケジューラは1秒毎に実行可能状態のプロセスを選択。

→ 単純な無限ループプログラムを2つ実行させてみる。

Basicプログラム：b.out

```
10 goto 10
```

UNIXコマンド

```
# echo run | bas b.out &
# echo run | bas b.out &
```

10	14	1	/etc/init	2	10	wait	L---	0204400-0211577(2.6KB)
11	15	1	-	62	10	wait	L---	0233500-0244177(4.3KB)
12	16	15	echo run	15	-50	zomb	L---	007674- 007703(8blk)
13	17	15	bas b.out	28	105	run	L---	0267500-0316277(11.4KB)
14	18	15	echo run	11	-50	zomb	L---	007704- 007713(8blk)
15	19	15	bas b.out	13	102	run	L---	0331000-0357577(11.4KB)
16	0	0	none	0	0	free		00000000-00000000(0:PKD)

(a) プロセス構造体リスト画面

15	19	15	bas b.out	15	104	run	L---	0331000-0357577(11.4KB)	35.018320
13	17	15	bas b.out	16	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	17	105	run	L---	0331000-0357577(11.4KB)	35.018320
13	17	15	bas b.out	18	105	run	L---	0267500-0316277(11.4KB)	35.018320
4	6	1	/etc/update	19	105	run	L---	0331000-0357577(11.4KB)	35.018320
4	6	1	/etc/update	20	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	21	105	run	L---	0331000-0357577(11.4KB)	35.018320
15	19	15	bas b.out	22	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	23	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	24	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	25	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	26	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	27	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	28	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	29	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	30	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	31	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	32	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	33	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	34	105	run	L---	0267500-0316277(11.4KB)	35.018320
13	17	15	bas b.out	35	105	run	L---	0267500-0316277(11.4KB)	35.018320
15	19	15	bas b.out	36	105	run	L---	0267500-0316277(11.4KB)	35.018320

(b) コンテキストスイッチ履歴画面



【事例③】メモリ不足時のプロセス再配置

スタック領域を 48KB 以上消費する
プログラムを実行。

Cプログラム : test1.c

```
int pid;

main(){
    int x[8192]; /* 16KB */
    int y[8192]; /* 16KB */
    int z[8192]; /* 16KB */
    pid=getpid();
    printf("pid:%d %o(8), addr:%o(8)\n",
        pid,pid,&pid);
    while(1);
}
```

UNIX コマンド

```
# cc test1.c
# ./a.out
```

pid	ppid	cmd	cx	pri	stat	flag	mem(oct)
0	0	##KERNEL##					# 0000000-0116077(39.1KB)
0	0	[sched]	24	-100	slep LS--	0116100-0120077(1.0KB)	* 0123500-0125477(1.0KB)
1	1	**FREE MEMORY**					
1	1	/etc/init	20	40	wait L---	0125500-0132677(2.6KB)	
2	7	/etc/init	4	10	wait L---	0132700-0140077(2.6KB)	
3	8	/etc/init	2	10	wait L---	0140100-0145277(2.6KB)	
5	9	/etc/init	2	10	wait L---	0145300-0152477(2.6KB)	
6	10	/etc/init	2	10	wait L---	0152500-0157677(2.6KB)	
7	11	/etc/init	2	10	wait L---	0157700-0165077(2.6KB)	
		FREE MEMORY					* 0165100-0165277(0.1KB)
4	6	/etc/update	3	90	wait L---	0165300-0171777(2.3KB)	
8	12	/etc/init	2	10	wait L---	0172000-0177177(2.6KB)	
9	13	/etc/init	2	10	wait L---	0177200-0204377(2.6KB)	
10	14	/etc/init	2	10	wait L---	0204400-0211577(2.6KB)	
		FREE MEMORY					* 0223400-0233477(4.1KB)
11	15	1 -	62	100	run L---	0233500-0244177(4.3KB)	
		FREE MEMORY					* 0244200-0377777(45.9KB)
		##RESERVED##					# 0400000-0757777(120.0KB)
		##I/O##					# 0760000-0777777(8.0KB)

現在の最大連続空きメモリ領域 (46KB)
(a)a.out起動前

pid	ppid	cmd	cx	pri	stat	flag	mem(oct)
0	0	##KERNEL##					# 0000000-0116077(39.1KB)
0	0	[sched]	51	-100	slep LS--	0116100-0120077(1.0KB)	
12	22	./a.out	59	101	run L---	0120100-0266577(51.3KB)	
		FREE MEMORY					* 0266600-0377777(36.6KB)
		##RESERVED##					# 0400000-0757777(120.0KB)
		##I/O##					# 0760000-0777777(8.0KB)

システムプロセス(PID:0)と、a.out(PID:22)だけがオンメモリ
(b)a.out起動後

【事例④】ゾンビプロセスの発生、 親プロセスによるリソース開放



echo と bas をバックグラウンドで実行する。

UNIXコマンド

```
# echo &  
# bas &
```

10	14	1 /etc/init	2	10	wait L---	0204400-0211577(2.6KB)
11	15	1 -	58	10	wait L---	0233500-0244177(4.3KB)
12	16	15 echo	12	-50	zomb L---	007656- 007665(8blk)
13	17	15 bas	12	-50	zomb L---	007666- 007675(8blk)
14	0	0 none	0	0	free ---	0000000-0000000(0.0KB)
15	0	0 none	0	0	free ---	0000000-0000000(0.0KB)
16	0	0 none	0	0	free ---	0000000-0000000(0.0KB)
17	0					
18	0					
19	0					
20	0					
21	0					
22	0					
23	0					
24	0					
25	0					
26	0					
27	0	0 none	0	0	free ---	0000000-0000000(0.0KB)

- シェル(PID:15)は端末入力を待っている状態。
- echo と bas (PID:16と17) は終了してゾンビ状態。シェルはバックグラウンドで起動したプログラムの終了を待たないためゾンビ状態のままになる。

(a)ゾンビプロセスの発生

11	15	1 -	62	40	wait L---	0233500-0244177(4.3KB)
12	0	0 none	0	-50	free ---	0000000-0000000(0.0KB)
13	0	0 none	0	-50	free ---	0000000-0000000(0.0KB)
14	18	15 cal 2024	39	101	run L---	0244200-0253077(3.4KB)
15	0	0 none	0	0	free ---	0000000-0000000(0.0KB)
16	0	0 none	0	0	free ---	0000000-0000000(0.0KB)
17	0					
18	0					
19	0					
20	0					
21	0					
22	0	0 none	0	0	free ---	0000000-0000000(0.0KB)

シェルでフォアグラウンドでプログラムを実行させると子プロセス終了待ちを行うため、そこでゾンビを開放

(b)ゾンビプロセスの開放

5. むすび

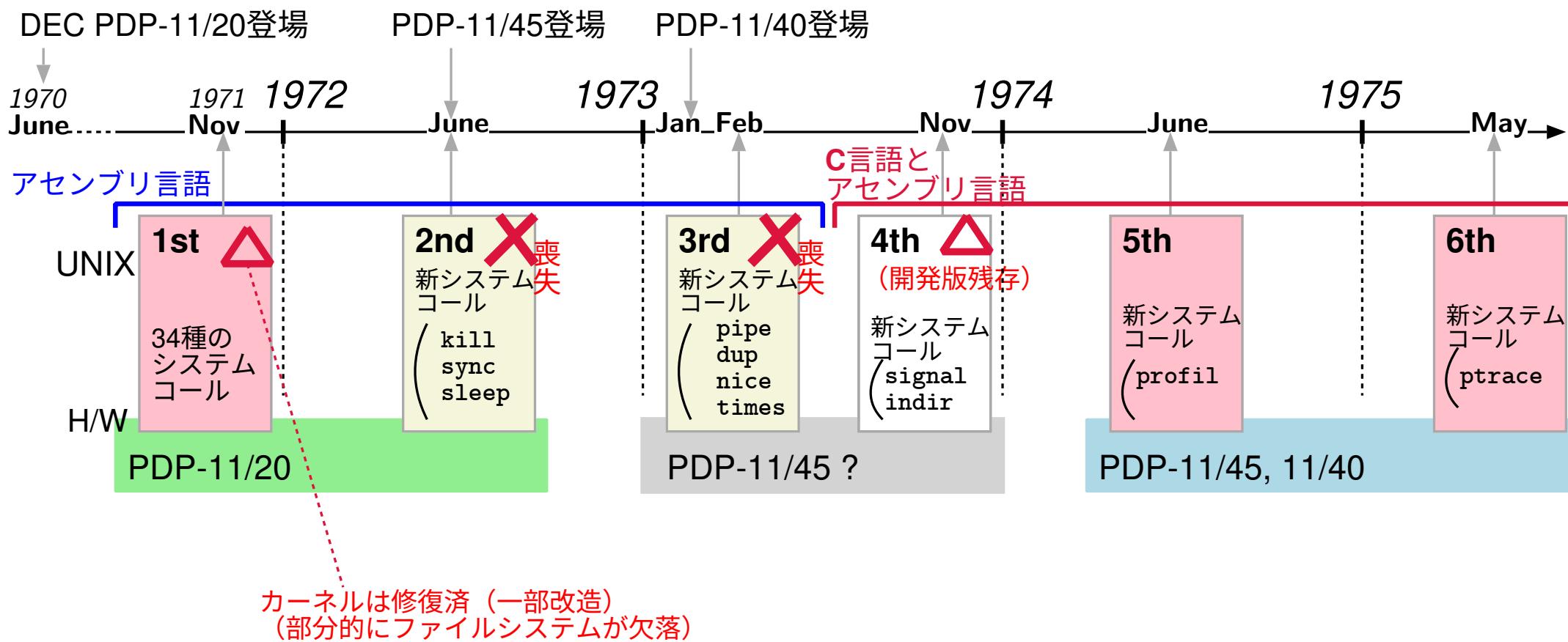
- UNIX第5版の解析用に開発したトレーサを改良し，テキスト画面による詳細な情報を表示できるようにした。
- プロセス構造体やユーザ構造体に含まれないコマンド名や，コンテクストスイッチ回数を取得・表示できるようにした。
- 新トレーサにより，コンテクストスイッチの様子や，メモリの使用状況，スワップアウト先などの情報を把握しやすくなったことを事例によって示した。
- 今後の課題
 - 不具合の修正（プロセス状態が正しくないことがある）
 - メモリレイアウト画面への「共有テキスト領域」の追記，スワップレイアウト画面の作成，システムコール画面の改善
 - 踏み込んだ解析（高負荷時のスケジューリング，スラッシング）
 - UNIX第6版，第7版への対応

謝辞 本研究はJSPS科研費21K00256助成を受けた

文献

- [1] 工藤信一朗, 鷹合大輔, 田村修, 石井充, ”**UNIX第1版におけるタスク切り替え機構の解析**”, 信学技法 (FIIS-22-558), 2022.
- [2] 鷹合大輔, 田村修, 工藤信一朗, ”**初期 UNIX におけるプロセス切り替え機構の動的解析**”, 信学技法 (FIIS-23-570), 2023.
- [3] K.Thompson, D.M.Ritchie, ”**The UNIX Time-Sharing System**”, Communications of the ACM, Vol.17, No.7, 1974.
- [4] simh, ”**The Computer History Simulation Project**”, <https://github.com/simh/simh>, 参照 Mar.14, 2022.

【参考】 リサーチUNIXの状況



【参考】 UNIX第1版と第5版の比較

	UNIX第1版(1971)	UNIX第5版(1974)
ミニコンピュータ	PDP-11/20 (1970)	PDP-11/45 (1972) PDP-11/40 (1973)
CPU動作モード	無し	ユーザ-/カーネル/SVモード
仮想記憶	×	○
メインメモリ(最大)	56KB	248KB
カーネル空間	16KB	64KB
ユーザ空間	8KB (修復版16KB)	64KB
管理プロセス数	16	50
オンメモリプロセス数	1	複数可
システムコール数	34	40
シグナル	×	○
パイプ	×	○
マルチユーザ	○	○
プロセス切り替え間隔	0.5秒	1秒

カーネルは起動時の最終処理として、スケジューラ（システムプロセス）と、initプロセス（/etc/rcの実行、ログイン準備など）を立ち上げる。

/usr/sys/ken/main.c—— (カーネルソース)

```
:  
main()  
{  
:  
「スケジューラ (=システムプロセス, PID:0)」の準備  
proc[0].p_addr = *ka6;  
proc[0].p_size = USIZE;  
proc[0].p_stat = SRUN;  
proc[0].p_flag = | SLOAD|SSYS;  
u.u_procp = &proc[0];  
  
:  
          プロセス(PID:1)の生成と  
          initプログラムの起動準備  
if(newproc()) {  
    expand(USIZE+1);  
    u.u_uisa[0] = USIZE;  
    u.u_uisd[0] = 6;  
    sureg();  
    copyout(icode, 0, 30);  
    return;  
}  
sched();           スケジューラ開始
```

/usr/source/s1/init.c—— (initのソース※)

※わかりやすいように定数マクロを一部展開

```
:  
main()  
{  
:  
          子プロセスを生成  
i = fork();  
if(i == 0) {  
    open("/", 0);  
    dup(0);  
    dup(0);  
    execl("/bin/sh",  
          "/bin/sh", "/etc/rc", 0);  
    exit();  
}  
while(wait() != i);  
:  
          「端末リスト」のオープン  
fi = open("/etc/ttys", 0);  
:  
for(all)  
    if(p->line != 0 && p->pid == 0)  
        dfork(p);  
for(ever){  
    子プロセスを生成する。子プロセス側で  
    端末デバイスをオープンした後、gettyを  
    実行 (ログインプロンプトの表示).  
}
```

