Supplementary data

This is the supplementary material for the paper "No Crash, No Exploit: Automated Verification of Embedded Kernels". We provide additional data and explanations for the experimental evaluations RQ2 and RQ3.

1 Evaluation of the methodology (RQ2)

Protocol. The goal of this experiment is to evaluate our 3-step methodology, in particular (1) whether our shape domain is needed, (2) what is the nature and impact of the shape annotations, and (3) whether differentiated handling of boot code is mandatory.

We experiment on the **v1** kernel version, and report results for both the boot code and the runtime, using different sets of annotations with an increasing amount of annotations:

- *No annotation* (equivalent to having no shape domain);
- *Generated* annotations (without any manual annotations);
- *Minimal* annotations with which the analysis terminate;
- *Generic* and *Specific* are as defined in the core article;
- *Dedicated* hardcodes some parameters, such as the number of tasks, with values of the sample user tasks.

Table 1: Impact of the methodology.

Annotations	No annotation		Generated		Minimal		Generic		Specific		Dedicated	
Generated annotations (total)	0		1057		1057		1057		1057		1057	
Manual annotations (total)	0		0		10		57		58		62	
	boot	runtime	boot	runtime	boot	runtime	boot	runtime	boot	runtime	boot	runtime
Analysis time (s)	✗	N/A	✗	N/A	342	394	195	222	187	219	151	203
Total number of alarms	N/A	N/A	N/A	N/A	85	13	60	1	59	0	43	0
User tasks checking	N/A		N/A		✓		✓		✓		✓	

Results. Table 1 shows the result of this evaluation. The analysis does not succeed in finding an invariant without the shape domain or without manual annotations — the analysis is too imprecise and aborts in boot, denoted ✗. Only 10 lines of manual annotations are necessary for the analysis to complete (***minimal***), albeit with many alarms in both boot code and runtime. These annotations mainly limit the range of array indices to prevent overflows in pointer arithmetic. The ***generic*** configuration adds 47 lines indicating which pointer types or structure fields may be null, which fields hold array indices, and relating array lengths with memory locations holding these lengths. This configuration eliminates most alarms in the runtime, but 60 alarms remain in the boot code. The ***specific***

annotations reach 0 alarms in runtime, but still 59 alarms in boot code. Even the ***dedicated*** annotations cannot eliminate all alarms in boot code.

Interestingly, we also found that some of the invariants in the generated annotations do not hold before boot.

Conclusions. *Parameterized verification of the kernel cannot be done without the shape domain.* The ability to extract the annotations from interface types is extremely useful, as *95% of the annotations are automatically extracted*, requiring only 57 lines of simple manual annotations. Finally, *differentiated handling of boot code is necessary* as both the boot code is much harder to analyze than the runtime, and the shape invariants holds only after boot code terminates.

2 Measurements for the 96 EducRTOS variants (RQ3)

We give here the detailed data for the analysis of all 96 variants of EducRTOS. The instruction count is the number of (unique) instructions present in the CFG computed by the analysis.

Compiler	Opt. flag	Scheduling	Dyn. threads	Printing	Time (s)	#instr.	Max. mem. (MB)
gcc	-O1	RR	yes	no	1.6	2649	283
gcc	-O1	RR	yes	yes	40.5	2734	5704
gcc	-O1	RR	no	no	1.3	2640	253
gcc	-O1	RR	no	yes	40.5	2725	5671
gcc	-O1	FP	yes	no	7.1	2695	672
gcc	-O1	FP	yes	yes	72.6	2780	9287
gcc	-O1	FP	no	no	5.5	2686	559
gcc	-O1	FP	no	yes	70.2	2771	9072
gcc	-O1	EDF	yes	no	8.3	2711	753
gcc	-O1	EDF	yes	yes	72.3	2796	9185
gcc	-O1	EDF	no	no	6.3	2702	602
gcc	-O1	EDF	no	yes	70.9	2787	9259
clang	-O1	RR	yes	no	1.8	2400	325
clang	-O1	RR	yes	yes	26.9	2501	3608
clang	-O1	RR	no	no	1.5	2387	280
clang	-O1	RR	no	yes	26.3	2488	3598
clang	-O1	FP	yes	no	4.1	2476	518
clang	-O1	FP	yes	yes	28.9	2577	3792
clang	-O1	FP	no	no	4.7	2463	538
clang	-O1	FP	no	yes	45.5	2564	5696
clang	-O1	EDF	yes	no	4.6	2484	517
clang	-O1	EDF	yes	yes	29.5	2585	3802
clang	-O1	EDF	no	no	5.4	2471	575
clang	-O1	EDF	no	yes	46	2572	5689
gcc	-O2	RR	yes	no	1.7	2705	278
gcc	-O2	RR	yes	yes	41	2787	5705
gcc	-O2	RR	no	no	1.4	2642	258
gcc	-O2	RR	no	yes	40.1	2724	5695
gcc	-O2	FP	yes	no	7	2750	669

gcc	-O2	FP	yes	yes	72.8	2832	9091
gcc	-O2	FP	no	no	5.4	2687	555
gcc	-O2	FP	no	yes	70.5	2769	9120
gcc	-O2	EDF	yes	no	8.2	2766	726
gcc	-O2	EDF	yes	yes	73.3	2848	9108
gcc	-O2	EDF	no	no	6.3	2703	586
gcc	-O2	EDF	no	yes	70.9	2785	9168
clang	-O2	RR	yes	no	1.7	2361	299
clang	-O2	RR	yes	yes	26.4	2429	3566
clang	-O2	RR	no	no	1.3	2423	264
clang	-O2	RR	no	yes	25.8	2491	3557
clang	-O2	FP	yes	no	4	2429	472
clang	-O2	FP	yes	yes	28.6	2497	3737
clang	-O2	FP	no	no	4.5	2490	515
clang	-O2	FP	no	yes	44.8	2558	5704
clang	-O2	EDF	yes	no	4.5	2436	488
clang	-O2	EDF	yes	yes	29	2504	3747
clang	-O2	EDF	no	no	5.1	2498	537
clang	-O2	EDF	no	yes	45.4	2566	5715
gcc	-O3	RR	yes	no	1.7	2705	288
gcc	-O3	RR	yes	yes	29.9	2796	4232
gcc	-O3	RR	no	no	1.4	2642	258
gcc	-O3	RR	no	yes	29.2	2733	4281
gcc	-O3	FP	yes	no	7	2739	647
gcc	-O3	FP	yes	yes	52.5	2830	6805
gcc	-O3	FP	no	no	5.4	2676	550
gcc	-O3	FP	no	yes	51.2	2767	6795
gcc	-O3	EDF	yes	no	8.5	2761	742
gcc	-O3	EDF	yes	yes	53.1	2852	6850
gcc	-O3	EDF	no	no	6.2	2698	592
gcc	-O3	EDF	no	yes	51.9	2789	6678
clang	-O3	RR	yes	no	1.7	2486	300
clang	-O3	RR	yes	yes	25.4	2788	3516
clang	-O3	RR	no	no	1.4	2423	264
clang	-O3	RR	no	yes	25.2	2725	3401
clang	-O3	FP	yes	no	4	2556	470
clang	-O3	FP	yes	yes	45	2858	5691
clang	-O3	FP	no	no	4.6	2492	523
clang	-O3	FP	no	yes	43.4	2794	5419
clang	-O3	EDF	yes	no	4.6	2564	508
clang	-O3	EDF	yes	yes	45.5	2866	5658
clang	-O3	EDF	no	no	5.2	2501	545
clang	-O3	EDF	no	yes	44.5	2803	5438
gcc	-Os	RR	yes	no	1.7	2664	275
gcc	-Os	RR	yes	yes	32.4	2750	4612
gcc	-Os	RR	no	no	1.3	2652	244

gcc	-Os	RR	no	yes	31.8	2738	4514
gcc	-Os	FP	yes	no	6.6	2717	628
gcc	-Os	FP	yes	yes	55.1	2803	7063
gcc	-Os	FP	no	no	5.1	2705	529
gcc	-Os	FP	no	yes	54	2791	6930
gcc	-Os	EDF	yes	no	7.8	2730	699
gcc	-Os	EDF	yes	yes	57.1	2816	7302
gcc	-Os	EDF	no	no	5.8	2718	567
gcc	-Os	EDF	no	yes	54.2	2804	7062
clang	-Os	RR	yes	no	1.7	2368	312
clang	-Os	RR	yes	yes	20.7	2439	2864
clang	-Os	RR	no	no	1.4	2346	268
clang	-Os	RR	no	yes	20.3	2417	2841
clang	-Os	FP	yes	no	4.1	2437	484
clang	-Os	FP	yes	yes	23.1	2508	3046
clang	-Os	FP	no	no	4.5	2413	489
clang	-Os	FP	no	yes	34.2	2484	4320
clang	-Os	EDF	yes	no	4.7	2445	522
clang	-Os	EDF	yes	yes	23.7	2516	3022
clang	-Os	EDF	no	no	5.2	2423	552
clang	-Os	EDF	no	yes	34.9	2494	4408