Lab 1 – Dynamic Power Management

1. Objective

The lab is devoted to the understanding of the basics of DPM by using and modifying a simple power state machine simulator in C for the evaluation of power management policies.
The lab will cover 6 hours overall (1.5+3+1.5) as follows:

Day 1	Compile and run the DPM simulator
	Workload profile generation
Days 2/3	Timeout policy extension
	Implementation of history-based predictive policy

The assignment will consist of three parts; however, **a single report will have to be delivered. This lab will have a maximum score of 2 points.**

2. The DPM Simulator

This is a C program with the following basic operations

- Read a power model in terms of a PSM
- Set the parameters according to the PSM values
- Read a workload profile
- Simulate two power management policies
  - Timeout
  - History-based prediction

Usage

`dpm_simulator [-t|-h] [-psm <PSM file>] [-wl <wl file>]`
`-t <Timeout>`: use a timeout policy with `<Timeout>`
`-h <Value1> …<Value5> <Threshold1> <Threshold2>:` use a history-based predictive policy (length of history window =5)
- <Value1-5> coefficient values
- <Threshold1-2> time thresholds

`-psm <psm file>:` the power state machine (PSM) definition file name
`-wl <wl file>:` the workload file name

Power state machine (PSM) file format

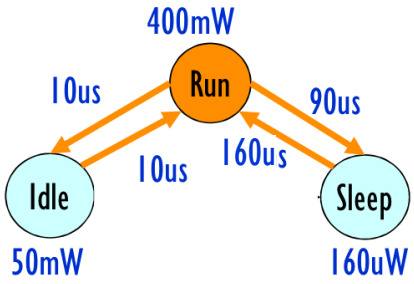
PSMs used by the simulator must be specified as text files with the following format.

- Default time, power, and energy units are us, mW and uJ.
- The first file row contains the **power** a state in each column (P(S0) P(S1), …, P(SN)).
- Subsequent rows contain **transitions costs** in the format: **<energy>/<time>**.
- The element in row i+1, and column j corresponds to the transition costs between states Si and Sj
- Transitions marked as 0/0 are self loops (e.g., from RUN to RUN), while transactions marked as -1/-1 are not supported by the PSM (e.g., from SLEEP to IDLE).

Note that the provided simulator only supports PSMs with three states, but the format and code can be easily extended to support more complex PSMs.

Example of PSM specification:

	(RUN)	(IDLE)	(SLEEP)
	400	50	0.16
(RUN)	0/0	10/10	10/90
(IDLE)	10/10	0/0	-1/-1
(SLEEP)	30/160	-1/-1	0/0

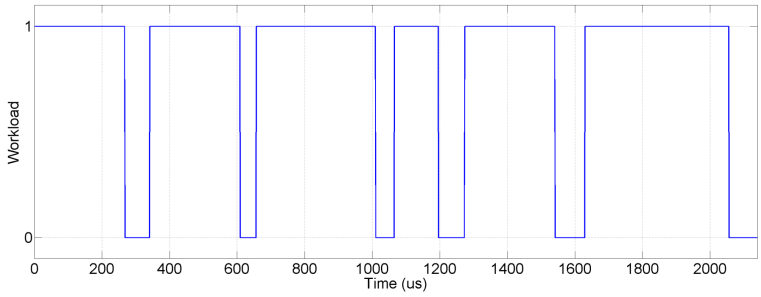


* Grey text not in the file

Workload format

The workload is given as a list of idle intervals. Values are in us. Example (wl_example.txt):

268	341	*start of 1st idle period, end of 1st idle period*
609	656	*start of 2nd idle period, end of 2nd idle period*
1010	1065	.
1196	1273	.
1541	1629	.
2056	2139	.



Sample output of the tool

Based on timeout policy (timeout = 20us) with `example/wl.txt` as workload file.

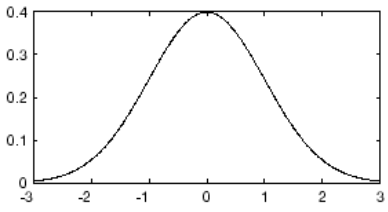
```
➜ dpm_simulator git:(master) ✗ ./dpm_simulator -t 20 -psm example/psm.txt -wl example/wl.txt
[psm] State Run: power = 400.00mW
[psm] State Idle: power = 50.00mW
[psm] State Sleep: power = 0.16mW
[psm] Run -> Idle transition: energy  = 10uJ, time = 10us
[psm] Run -> Sleep transition: energy  = 10uJ, time = 90us
[psm] Idle -> Run transition: energy  = 10uJ, time = 10us
[psm] Sleep -> Run transition: energy  = 30uJ, time = 160us

[sim] Active time in profile = 0.300130s
[sim] Idle time in profile = 0.244066s
[sim] Total time = 0.544196s
[sim] Timeout waiting time = 0.024679s
[sim] Total time in state Run = 0.324809s
[sim] Total time in state Idle = 0.219387s
[sim] Total time in state Sleep = 0.000000s
[sim] Time overhead for transition = 0.022770s
[sim] N. of transitions = 2277
[sim] Energy for transitions = 0.022770J
[sim] Energy w/o DPM = 0.217678J, Energy w DPM = 0.163663J
[sim] 24.8 percent of energy saved.
```

3. Detailed Lab plan

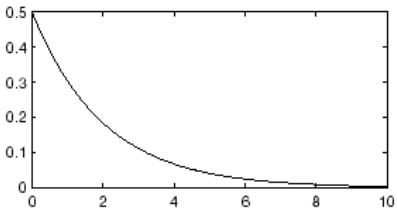
Part 1: Workload profile generation (Day 1)
We want to generate workload profiles distributed as follows:

- Active periods (Uniform distribution, min = 1us, max = 500us), at least 5000 samples
- Idle periods distributed in various ways (see also figures below)
    - Uniform distribution, min = 1us, max=100us (high utilization)
    - Uniform distribution, min=1us, max=400us (low utilization)
    - Normal distribution, mean=100us, standard deviation=20
    - Exponential distribution, mean=50us
    - Tri-modal distribution, mean = 50, 100, 150us, standard deviation=10: obtained as combination of 3 normal distributions

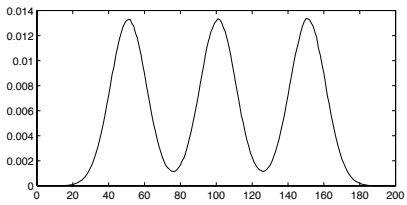
We will also provide you with two additional workloads of unknown distribution.



Standard normal PDF, with $\mu = 0$ and $\sigma = 1$



Exponential PDF, with $\mu = 2$



Trimodal PDF, with $\mu = 50$, 100, 150 and $\sigma = 20$

Part 2: Extension of timeout policy (Days 2/3)
Modify the simulator enabling the transition to the SLEEP state with a timeout policy (the provided simulator only supports the transition from RUN to IDLE).
Main functions to modify:

- `dpm_decide_state()` from `dpm_policies.h/.c`
- `parse_args()` from `utilities.h/.c`

Part 3: history-based predictive policy implementation (Days 2/3)
Modify the simulator adding the implementation of an history-based predictive policy. The simulator is already designed to receive from the command line the parameters of the history-based policy. Specifically, it accepts:

- five regression coefficients for history-based idle interval prediction
- two thresholds corresponding to the minimum predicted intervals that trigger a transition from RUN to IDLE and to SLEEP respectively.

However, the policy itself is not implemented.
Main functions to modify:

- `dpm_decide_state()` from `dpm_policies.h/.c`
- `parse_args()` from `utilities.h/.c` (e.g. to change the number of coefficients)

4. Report assignments

Assignment 1 report:
- Description of generated workload profiles

Assignment 2 report:
- Result of timeout policy with the generated and provided profiles
- Timeout vs. energy saving (sweep timeout from 1us to the maximum length of idle time)
- Comparison between example timeout and extended timeout (with RUN-> SLEEP transition)
- Analysis on timing and energy overhead

Assignment 3 report:
- Description of implemented predictive policy
- Result of implemented predictive policy with the profiles
- Analysis on window size vs. energy saving
- Analysis on coefficient values vs. energy saving (model order)
- Comparison between predictive and timeout policies
- Analysis on timing and energy overhead

A single report needs to be delivered for the whole lab.