OS Project 1 Report

資工三 B06902011 陳義誠

Design

- The program is designed for a 2(or more)-CPU system. The scheduler would run on CPU 0 and would assgin/halt/restart scheduled process on CPU 1. To fulfill this **sched_setaffinity()** is used to keep the processes on the same CPU.
- **sched_setscheduler()** with specific sched_priority in sched_param to set the priority of the processes, making them like running or being replaced.
- For FIFO, just run the process that comes earlier and check if there is any process ready when idle.
- For RR, implement a ready queue where every ready process would be pushed. When a process's time quantum is due and it does not finish executing, re-push it into the queue and pop 1 from the queue. If a process is re-pushed and another process arrives at the same time, the arriving one would be pushed first.
- For SJF, each time CPU 1 is available or a process terminates, choose a ready process (if there is) with shortest execution time.
- For PSJF, do the check described above each time unit.

The program consists of the following parts:

main.c

The main program to execute and fulfill the requirments. It would execute the following in order:

- Scan the input and initialize the enviornment, including preparing the data, moving the scheduler to the first CPU, and setting the scheduler with maximum priority.
- For every time unit, do:
  - If the a running process has been running for its execution time, wait for its termination.
  - If a process arrives, call **fork()** and make it ready to be executed (not necessarily immediately).
    For parent (scheduler), do:
    - Select the next process to run with specific scheduling policy.
    - If changing processes is needed, do context switch by changing the currently-running process's priority as minium, and change the process running next with highest priority.

        - If a process i running, count as if it runs for a time unit.
    - Do the loop until all the processes terminates.
    - As for child, run its execution time after being forked and then sent on CPU 1. At exit, it would print its process name and pid on stdout, and the pid, start time, and end time to dmesg.

scheduler.c

Including functions that define how to choose the next running process under specific scheduling policy.

queue.c

The implementation of the ready queue or RR. FIFO doesn't use the queue since the processes are in the order of the ready time.

functions.c

Other utilization such as moving processes on specific CPU, setting processes' priority, execution of a process, and so on.

process_time.c

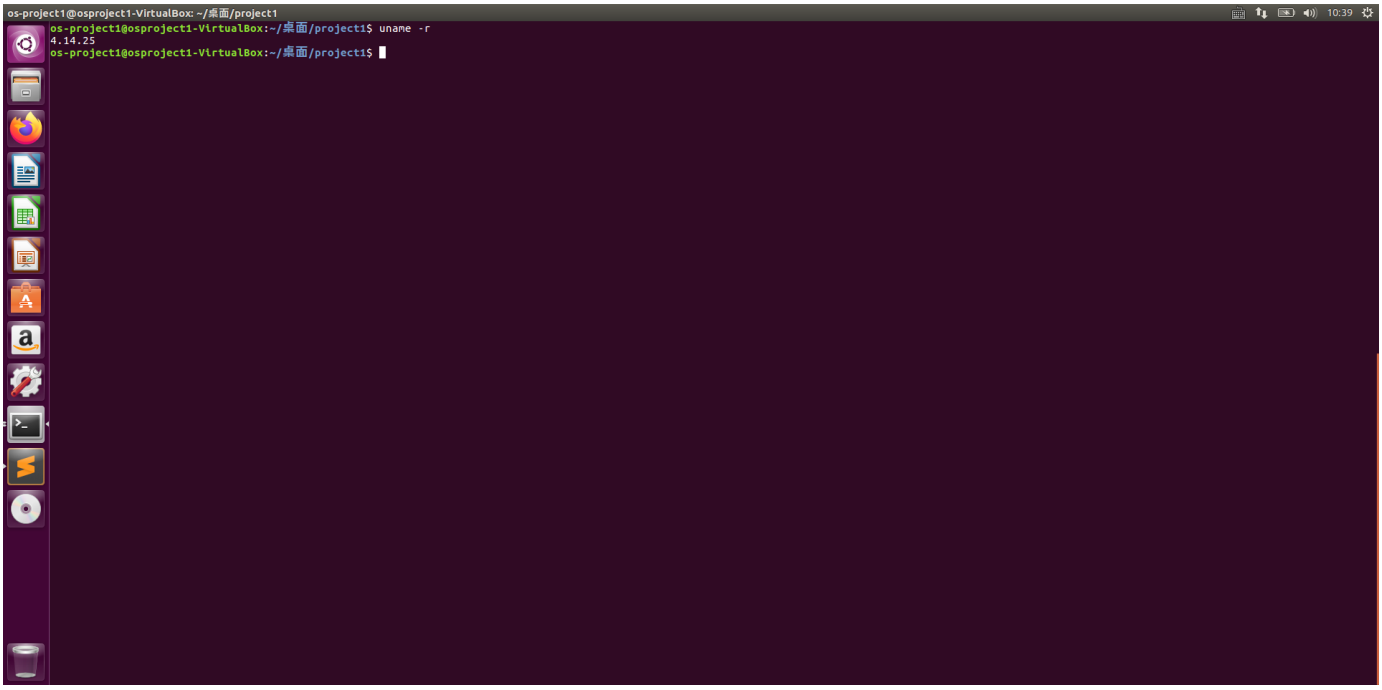
Implement the self-defined system call function **sys_process_time()**. User would get the time with **getnstimeofday()**. If a flag tolds it that this is the end of a process, it would print the results of the execution time in dmseg with **printk**().

project1.h

including self-defined structures, some macros, and the prototype of the self-defined functions.

Kernel Version

The kernel version used is 4.14.25



Error between Theorem and Practice

The error between theoretical and pratical exection time is as follows. The error rate is |pratical_exec_time−expected_exec_time|/expected_exec_time.

- The unit time is 0.001866.

FIFO_1.txt

process	expected start time	expected end time	expected exec time	pratical start time	pratical end time	pratical exec time	err rat
P1	0	500	500	0.00	500.93	500.93	0.1
P2	500	1000	500	505.01	1017.72	512.71	2.5
P3	1000	1500	500	1017.86	1530.22	512.36	2.4
P4	1500	2000	500	1532.01	2034.76	502.74	0.5
P5	2000	2500	500	2036.48	2521.85	485.37	2.9

FIFO_2.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	0	80000	80000	0.00	79197.00	79197.00	1.00%
P2	80000	85000	5000	80750.54	85957.74	5207.20	4.14%
P3	85000	86000	1000	85979.55	87002.74	1023.18	2.32%
P4	86000	87000	1000	87008.05	88034.51	1026.46	2.65%

FIFO_3.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	0	8000	8000	0.00	8129.71	8129.71	1.62%
P2	8000	13000	5000	8196.65	13279.31	5082.66	1.65%
P3	13000	16000	3000	13337.15	16260.02	2922.86	2.57%
P4	16000	17000	1000	16385.63	17392.05	1006.42	0.64%
P5	17000	18000	1000	17407.19	18421.57	1014.38	1.44%
P6	18000	19000	1000	18436.91	19505.65	1068.73	6.87%
P7	19000	23000	4000	19507.99	23630.91	4122.92	3.07%

FIFO_4.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	0	2000	2000	0.00	1991.87	1991.87	0.41%
P2	2000	2500	500	2052.62	2552.33	499.71	0.06%
P3	2500	2700	200	2553.24	2746.77	193.53	3.23%
P4	2700	3200	500	2761.64	3254.78	493.14	1.37%

FIFO_5.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	0	8000	8000	0.00	8068.65	8068.65	0.86%
P2	8000	13000	5000	8226.02	13288.07	5062.05	1.24%
P3	13000	16000	3000	13387.33	16385.23	2997.90	0.07%
P4	16000	17000	1000	16428.91	17404.10	975.18	2.48%
P5	17000	18000	1000	17427.20	18436.08	1008.88	0.89%
P6	18000	19000	1000	18466.08	19481.91	1015.84	1.58%
P7	19000	23000	4000	19499.08	23545.33	4046.25	1.16%

PSJF_1.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P4	3000	6000	3000	3144.17	6198.00	3053.83	1.79%
P3	2000	10000	8000	2080.22	10312.04	8231.82	2.90%
P2	1000	16000	15000	1038.57	16392.65	15354.08	2.36%
P1	0	25000	25000	0.00	25170.59	25170.59	0.68%

PSJF_2.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P2	1000	2000	1000	1020.24	2009.77	989.53	1.05%
P1	0	4000	4000	0.00	4012.28	4012.28	0.31%
P4	5000	7000	2000	5107.99	7116.67	2008.68	0.43%
P5	7000	8000	1000	7132.39	8074.32	941.93	5.81%
P3	4000	11000	7000	4086.48	11089.17	7002.70	0.04%

PSJF_3.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P2	500	1000	500	506.87	998.12	491.25	1.75%
P3	1000	1500	500	1006.22	1492.22	486.00	2.80%
P4	1500	2000	500	1506.19	1998.20	492.01	1.60%
P1	0	3500	3500	0.00	3516.10	3516.10	0.46%

PSJF_4.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P3	100	1100	1000	110.91	1103.39	992.48	0.75%
P2	0	3000	3000	0.00	2998.33	2998.33	0.06%
P4	3000	7000	4000	2998.58	6754.46	3755.88	6.10%
P1	7000	14000	7000	6918.42	13750.44	6832.02	2.40%

PSJF_5.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	100	200	100	0.00	97.57	97.57	2.43%
P3	200	400	200	99.95	300.53	200.58	0.29%
P2	400	4400	4000	317.11	4346.07	4028.96	0.72%
P4	4400	8400	4000	4397.55	8232.85	3835.31	4.12%
P5	8400	15400	7000	8468.23	15452.36	6984.13	0.23%

RR_1.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	0	500	500	0.00	504.71	504.71	0.94%
P2	500	1000	500	507.19	1024.84	517.66	3.53%
P3	1000	1500	500	1024.97	1533.85	508.88	1.78%
P4	1500	2000	500	1545.13	2047.80	502.67	0.53%
P5	2000	2500	500	2059.54	2559.02	499.48	0.10%

RR_2.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	600	8100	7500	0.00	7567.60	7567.60	0.90%
P2	1100	9600	8500	514.27	9189.05	8674.79	2.06%

RR_3.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P3	4200	18200	14000	3031.89	17206.78	14174.88	1.25%
P1	1200	19700	18500	0.00	18753.35	18753.35	1.37%
P2	2700	20200	17500	1528.47	19278.51	17750.04	1.43%
P6	8200	28200	20000	7059.81	27284.38	20224.58	1.12%
P5	6700	30200	23500	5540.51	29375.33	23834.82	1.42%
P4	6200	31200	25000	5027.74	30378.22	25350.48	1.40%

RR_4.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P4	1500	5500	4000	1500.15	5530.36	4030.21	0.76%
P5	2000	6000	4000	1992.66	5991.03	3998.37	0.04%
P6	2500	6500	4000	2512.91	6497.52	3984.62	0.38%
P3	1000	14500	13500	998.97	14740.91	13741.94	1.79%
P7	3500	18500	15000	3519.08	18828.48	15309.40	2.06%
P2	500	20000	19500	496.53	20284.62	19788.09	1.48%
P1	0	23000	23000	0.00	23457.23	23457.23	1.99%

RR_5.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P4	1500	5500	4000	1529.91	5510.22	3980.30	0.49%
P5	2000	6000	4000	2029.14	6046.31	4017.17	0.43%
P6	3000	7000	4000	3027.87	7020.23	3992.36	0.19%
P3	1000	14500	13500	1011.00	14622.51	13611.51	0.83%
P7	3500	18500	15000	3533.38	18633.75	15100.37	0.67%
P2	500	20000	19500	501.38	20038.91	19537.53	0.19%
P1	0	23000	23000	0.00	23278.46	23278.46	1.21%

SJF_1.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P2	0	2000	2000	0.00	1986.82	1986.82	0.66%
P3	2000	3000	1000	1987.18	2956.67	969.49	3.05%
P4	3000	7000	4000	2956.76	6826.60	3869.84	3.25%
P1	7000	14000	7000	6868.45	13453.86	6585.40	5.92%

SJF_2.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	100	200	100	0.00	101.32	101.32	1.32%
P3	200	400	200	101.51	301.11	199.59	0.20%
P2	400	4400	4000	305.37	4361.83	4056.46	1.41%
P4	4400	8400	4000	4368.78	8180.39	3811.61	4.71%
P5	8400	15400	7000	8416.31	15213.06	6796.75	2.90%

SJF_3.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	100	3100	3000	0.00	3054.65	3054.65	1.82%
P4	3100	3110	10	3071.15	3080.73	9.57	4.28%
P5	3110	3120	10	3081.05	3090.79	9.73	2.65%
P6	3120	7120	4000	3090.91	7039.51	3948.60	1.28%
P7	7120	11120	4000	7113.99	11090.73	3976.74	0.58%
P2	11120	16120	5000	11138.66	16103.43	4964.77	0.70%
P3	16120	23120	7000	16177.47	23202.81	7025.34	0.36%
P8	23120	32120	9000	23288.56	32165.50	8876.93	1.37%

SJF_4.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	0	3000	3000	0.00	3063.08	3063.08	2.10%
P2	3000	4000	1000	3063.43	4039.99	976.56	2.34%
P3	4000	8000	4000	4058.80	8082.10	4023.29	0.58%
P5	8000	9000	1000	8129.98	9131.52	1001.54	0.15%
P4	9000	11000	2000	9131.60	11149.26	2017.66	0.88%

SJF_5.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P1	0	2000	2000	0.00	1916.91	1916.91	4.15%
P2	2000	2500	500	2008.70	2487.47	478.77	4.25%
P3	2500	3000	500	2538.88	3042.59	503.70	0.74%
P4	3000	3500	500	3063.48	3554.40	490.93	1.81%

TIME_MEASUREMENT.txt

process	start time	end time	expected exec time	pratical start time	pratical end time	pratical exec time	error rate
P0	0	500	500	0.00	476.97	476.97	4.61%
P1	1000	1500	500	959.25	1441.29	482.04	3.59%
P2	2000	2500	500	1926.52	2425.32	498.80	0.24%
P3	3000	3500	500	2917.01	3431.03	514.01	2.80%
P4	4000	4500	500	3916.85	4427.85	511.00	2.20%
P5	5000	5500	500	4899.54	5403.31	503.77	0.75%
P6	6000	6500	500	5897.61	6393.84	496.24	0.75%
P7	7000	7500	500	6856.20	7375.74	519.54	3.91%
P8	8000	8500	500	7872.61	8370.68	498.07	0.39%
P9	9000	9500	500	8838.83	9338.40	499.57	0.09%

The average error rate is about 1.72%.

- In a part of the cases the practical execution time is longer than the expected one. It is because of some overhead such as calling system calls and operations of loops.
- Some of the cases is on the contrary. Maybe this is because the timer didn't count part of the execution of the process. For example, the scheduler thinks that it has switched the processes, but the original one still runs a small part.
- By leveraging multi-CPUs, the scheduler and the process can run at the same time so that the pratical execution time won't increase significantly due to context switch and waiting. The error rate is improved within 2%.
- Ther interference from others such as I/O interupt (such as the one of a mouse or a keyboard), running of other processes (such as a browser) can influence the performance. The worst error rate I got is 3.99%, nearly 2 times.