

MTCS-102(P)

Advanced Architecture

# Practical Assignment

Reg. No.: 23361

---

## Introduction:

Here in this experiment , we have seen the behavior of CPUs in terms of process scheduling and dynamic frequency scaling. We explored how processes are distributed across multiple CPUs, how CPU affinity can be controlled, and how dynamic frequency scaling affects CPU performance and power consumption.

## Determining Number of CPUs:

We determined the number of CPUs on the system using the **lscpu** command. This revealed that there were four CPUs numbered from 0 to 3.

```
et2023@dmacs13-OptiPlex-9020:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
Address sizes:          39 bits physical, 48 bits virtual
CPU(s):                 4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                 60
Model name:             Intel(R) Core(TM) i5-4670 CPU @ 3.40GHz
Stepping:               3
CPU MHz:               1100.000
CPU max MHz:           3800.0000
CPU min MHz:           800.0000
BogoMIPS:               6784.32
Virtualization:         VT-x
L1d cache:             128 KiB
L1i cache:             128 KiB
L2 cache:               1 MiB
L3 cache:               6 MiB
NUMA node0 CPU(s):     0-3
```

---

## Observing Process Affinity:

To know the behavior of the cpu threads and the execution program, a code was run that contained a large loop which was run throughout this experiment.

We opened two terminals and ran a program named **a.out** on one terminal while monitoring process activity using the **top** command on the other terminal. The 'P' column in the **top** display indicated the process ID that was most recently run by each process.

```
08268082780828808298083080831808328083380834808358 top - 15:25:40 up 18 days, 19:14, 1 user, load average: 0.70, 0.82, 0.93
43808448084580846808478084880849808508085180852808 Tasks: 329 total, 2 running, 327 sleeping, 0 stopped, 0 zombie
80861808628086380864808658086680867808688086980870 %Cpu(s): 11.7 us, 3.8 sy, 0.1 ni, 84.1 id, 0.3 wa, 0.0 hi, 0.0 si, 0.0 st
87880879880880888180882808838088480885808868088780 MlB Mem : 15898.3 total, 2959.3 free, 7274.7 used, 5664.2 buff/cache
58089680897808988089980900809018090280903809048090 MlB Swap: 2048.0 total, 1230.0 free, 818.0 used, 6927.3 avail Mem
09138091480915809168091780918809198092080921809228
30809318093280933809348093580936809378093880939809
80948809498095809591809528095380954809558095680957
9658096680967809688096809780971809728097380974809
28098380984809858098680987809888098980990809918099
10008100181002810038100481005810068100781008810098
17810188101981020810218102281023810248102581026810
81035810368103781038810398104081041810428104381044
05281053810548105581056810578105881059810608106181
98107081071810728107381074810758107681077810788107
10878108881089810908109181092810938109481095810968
04811058110681107811088110981110811118111281113811
81122811238112481125811268112781128811298113081131
13981140811418114281143811448114581146811478114881
68115781158811598116081161811628116381164811658116
11748117581176811778117881179811808118181182811838
91811928119381194811958119681197811988119981200812
81209812108121181212812138121481215812168121781218
22681227812288122981230812318123281233812348123581
```

P	PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2	35671	et2023	20	0	974164	52952	31468	R	29.8	0.3	45:28.07	gnome-termi
3	3066363	et2023	20	0	2496	580	512	S	5.3	0.0	0:00.16	a.out
3	25320	et2023	20	0	2955380	121080	106084	S	4.0	0.7	1251:32	wpsoffice
0	25398	et2023	20	0	689588	93852	90548	S	4.0	0.6	1201:20	wps
0	25522	et2023	20	0	865828	135916	114976	S	4.0	0.8	1259:01	et
1	1720	et2023	20	0	4700688	316440	63844	S	3.3	1.9	527:08.28	gnome-shell
0	3050022	et2023	20	0	3061384	263872	117692	S	2.3	1.6	1:58.03	Isolated Web
0	3043960	et2023	20	0	32.4g	105608	71056	S	2.0	0.6	27:36.92	code
3	337095	et2023	20	0	6419624	1.0g	261684	S	1.3	6.7	493:53.00	firefox
1	3043985	et2023	20	0	1130.9g	278084	98256	S	1.3	1.7	25:28.55	code
0	1519	et2023	20	0	567148	146828	84860	S	0.7	0.9	273:01.33	Xorg
0	3063128	et2023	20	0	3092940	532844	129508	S	0.7	3.3	1:06.63	Isolated Web
0	1076	mysql	20	0	2381380	309168	11416	S	0.3	1.9	104:48.29	mysqld
2	1432	et2023	9	-11	5606204	20612	16212	S	0.3	0.1	252:43.42	pulseaudio
3	1617	et2023	20	0	394932	19212	5860	S	0.3	0.1	10:14.38	ibus-daemon
3	1660	et2023	20	0	163244	7396	6248	S	0.3	0.0	2:10.60	ibus-engine-
0	442290	et2023	20	0	3314824	291268	133544	S	0.3	1.8	38:50.90	Isolated Web

Image1

## Process Affinity:

We observed that the program **a.out** was being executed on different CPUs at different points in time. The 'P' column in the **top** output reflected the process ID that was most recently run by each process. For instance, we noticed that the process corresponding to **a.out** switched between different CPUs when ran at two different instance of the execution of the same code(Image1 and Image2).

```
495964959749598495994960496014960249603496044960549606496074 top - 15:25:40 up 18 days, 19:14, 1 user, load average: 0.72, 0.82, 0.93
96134961449615496164961749618496194962049621496224962349624496 Tasks: 330 total, 2 running, 328 sleeping, 0 stopped, 0 zombie
30496314963249633496344963549636496374963849639496404964149642 %Cpu(s): 26.6 us, 4.0 sy, 0.0 ni, 69.3 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st
496484964949650496514965249653496544965549656496574965849659496 MlB Mem : 15898.3 total, 2950.4 free, 7282.7 used, 5665.2 buff/cache
9654966496674966849669496704967149672496734967449675496764967 MlB Swap: 2048.0 total, 1230.0 free, 818.0 used, 6918.8 avail Mem
24968349684496854968649687496884968949690496914969249693496944
97004970149702497034970449705497064970749708497094971049711497
17497184971949720497214972249723497244972549726497274972849729
49735497364973749738497394974049741497424974349744497454974649
75249753497544975549756497574975849759497604976149762497634976
94977049771497724977349774497754977649777497784977949780497814
97874978849789497904979149792497934979449795497964979749798497
9498054980649807498084980949810498114981249813498144981549816
49822498234982449825498264982749828498294983049831498324983349
349834983549836498374983849839498404984149842498434984449845
54985749858498594986049861498624986349864498654986649867498684
98744987549876498774987849879498804988149882498834988449885498
81498924989349894498954989649897498984989949900499014990249903
49909499104991149912499134991449915499164991749918499194992049
92649927499284992949930499314993249933499344993549936499374993
34994449945499464994749948499494995049951499524995349954499554
```

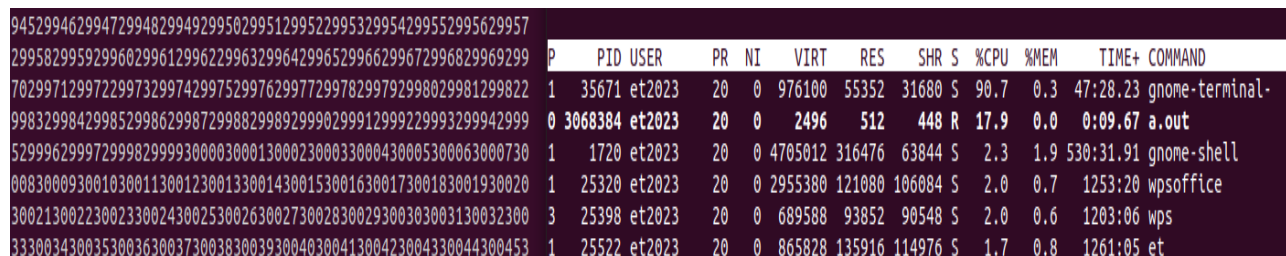
P	PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	35671	et2023	20	0	974308	53216	31468	R	87.7	0.3	45:32.90	gnome-termi
3	3066363	et2023	20	0	2496	580	512	S	14.3	0.0	0:01.02	a.out
0	1720	et2023	20	0	4700680	316428	63844	S	4.0	1.9	527:08.67	gnome-shell
0	25320	et2023	20	0	2955380	121080	106084	S	2.0	0.7	1251:32	wpsoffice
0	25522	et2023	20	0	865828	135916	114976	S	2.0	0.8	1259:01	et
0	25398	et2023	20	0	689588	93852	90548	S	1.7	0.6	1201:20	wps
3	337095	et2023	20	0	6419624	1.0g	261684	S	1.3	6.6	493:53.08	firefox
3	3050022	et2023	20	0	3061384	263872	117692	S	1.3	1.6	1:58.10	Isolated Web
3	1519	et2023	20	0	567760	146828	84860	S	1.0	0.9	273:01.39	Xorg
0	3043960	et2023	20	0	32.4g	105608	71056	S	1.0	0.6	27:36.98	code
0	3043985	et2023	20	0	1130.9g	279820	98256	S	1.0	1.7	25:28.61	code
1	3063570	root	20	0	0	0	0	I	1.0	0.0	0:02.43	kworker/u8:2+
3	3063609	root	20	0	0	0	0	I	1.0	0.0	0:01.92	kworker/u8:0+
3	43	root	20	0	0	0	0	S	0.7	0.0	6:03.48	kcompactd0

Image2

---

## Restricting CPU Affinity:

To restrict the affinity of a particular CPU, we used the `taskset` command. For example, running the command `taskset -c 0 ./a.out` limited the execution of `a.out` to CPU 0. This resulted in the **'P'** column consistently displaying 0 for the `a.out` process, indicating that it was consistently running on CPU 0(Image3).

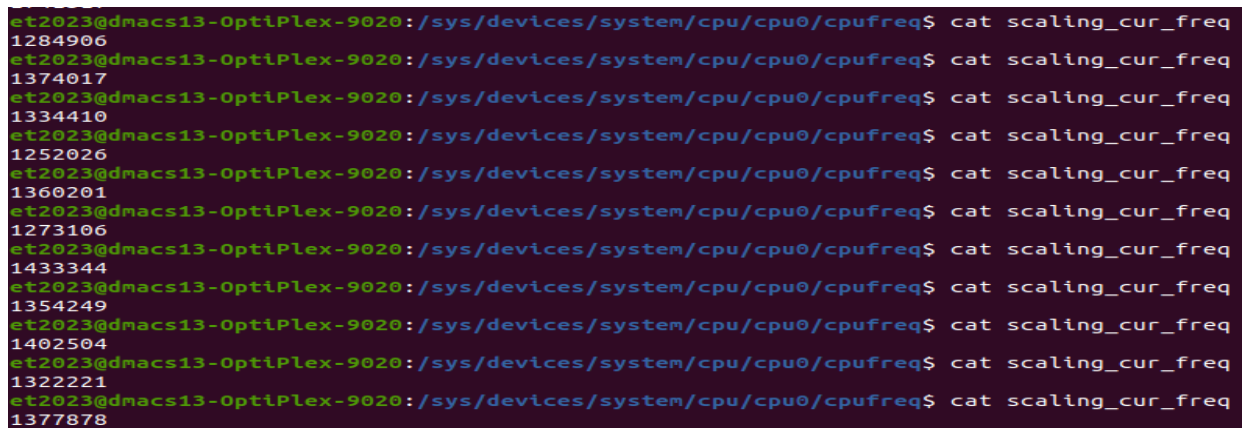


P	PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	35671	et2023	20	0	976100	55352	31680	S	90.7	0.3	47:28.23	gnome-terminal-
0	3068384	et2023	20	0	2496	512	448	R	17.9	0.0	0:09.67	a.out
1	1720	et2023	20	0	4705012	316476	63844	S	2.3	1.9	530:31.91	gnome-shell
1	25320	et2023	20	0	2955380	121080	106084	S	2.0	0.7	1253:20	wpsoffice
3	25398	et2023	20	0	689588	93852	90548	S	2.0	0.6	1203:06	wps
1	25522	et2023	20	0	865828	135916	114976	S	1.7	0.8	1261:05	et

Image3

## Dynamic Frequency Scaling:

Modern CPUs utilize dynamic frequency scaling to manage power consumption and heat generation. We investigated the dynamic nature of CPU frequency using the `/sys/devices/system/cpu/cpu0/cpufreq` file(Image4). This provided the current frequency of CPU 0 at various points in time. We found that the frequency wasn't fixed and changed based on the workload.



```
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1284906
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1374017
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1334410
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1252026
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1360201
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1273106
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1433344
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1354249
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1402504
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1322221
et2023@dmacs13-OptiPlex-9020: /sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1377878
```

Image4

---

The range of possible frequencies was determined using the **scaling\_min\_freq** and **scaling\_max\_freq** commands.

```
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_min_freq
800000
```

```
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_max_freq
3800000
```

### Conclusion:

This experiment provided insights into the behavior of CPUs, process scheduling, and dynamic frequency scaling. We observed how processes are distributed across CPUs, how CPU affinity can be controlled, and how dynamic frequency scaling influences CPU performance and energy consumption. These observations shed light on the efficient management of CPU resources and power consumption in modern computing systems.