National University of Computer & Emerging Sciences

Karachi Campus

PROJECT REPORT

Course Name: ***Operating Systems (CS2006)***

Project Name: ***Dining Philosophers***

Teacher Name: Miss Safia Baloch

Date: 26th May, 2022

GROUP MEMBERS:

Insha Samnani (20K-0247)

Ismail Ahmed Ansari (20K-0228)

Anjiya Muhammad Ali (20K-1687)

OBJECTIVE:

The problem is to **solve the deadlock** between 2 or more philosophers in the case if it occurs while eating (for the sake of chopsticks). The problem is how to design a discipline of behavior *(a concurrent algorithm)* such that no philosopher will starve, i.e., each can forever continue to alternate between eating and thinking, assuming that no philosopher knows when others may want to eat or think.

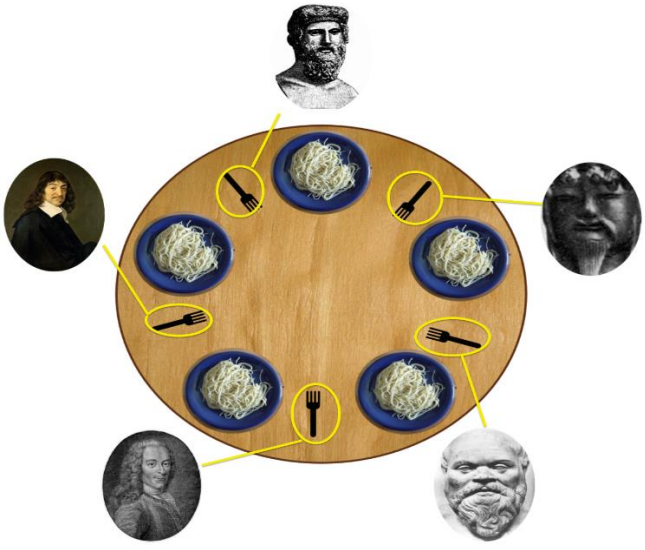
PROJECT DESCRIPTION:

INTRODUCTION:

Five silent philosophers sit at a round table with bowls of spaghetti. Forks are placed between each pair of adjacent philosophers as shown in the figure.

Each philosopher must alternately think and eat. However, a philosopher can only eat spaghetti when they have both left and right forks. Each fork can be held by only one philosopher and so a philosopher can use the fork only if it is not being used by another philosopher. After an individual philosopher finishes eating, they need to put down both forks so that the forks become available to others. A philosopher can only take the fork on their right or the one on their left as they become available and they cannot start eating before getting both forks.

Eating is not limited by the remaining amounts of spaghetti or stomach space; an infinite supply and an infinite demand are assumed.



METHODOLOGY:

- To initialize the philosophers eating we do the following:
- Number/identify each philosopher (Concept of ID)
- Initialize semaphore(s) to ensure each philosopher gets an opportunity to eat.
- Going clockwise (or maybe anti-clockwise) around the table each philosopher picks up the fork to his right.
- If it is unavailable, the code blocks the philosopher to take the right fork and puts in the waiting queue.
- Iterate again going clockwise and each philosopher who has access to his left & right fork starts eating.
- It does ***prevent deadlocks*** in each iteration (where N being the number of philosophers, that is equal to 5 in our case)

Configuration steps (Kernel Compile):

1. Download required packages
2. Open the terminal and use the wget command to download the Linux kernel source code: **wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.10.13.tar.xz**
3. When the file is ready, run the tar command to extract the source code**: tar xvf linux-4.10.13.tar.xz**
4. Make folder and save the main c code.
5. Save the **makefile**
6. Edit **syscalls.h** and **syscall_64.tlb** by adding our system call
7. Configure kernel by **make menuconfig**
8. Build the kernel by **make** command
9. Install required modules after compilation.

CODE & RESULTS:

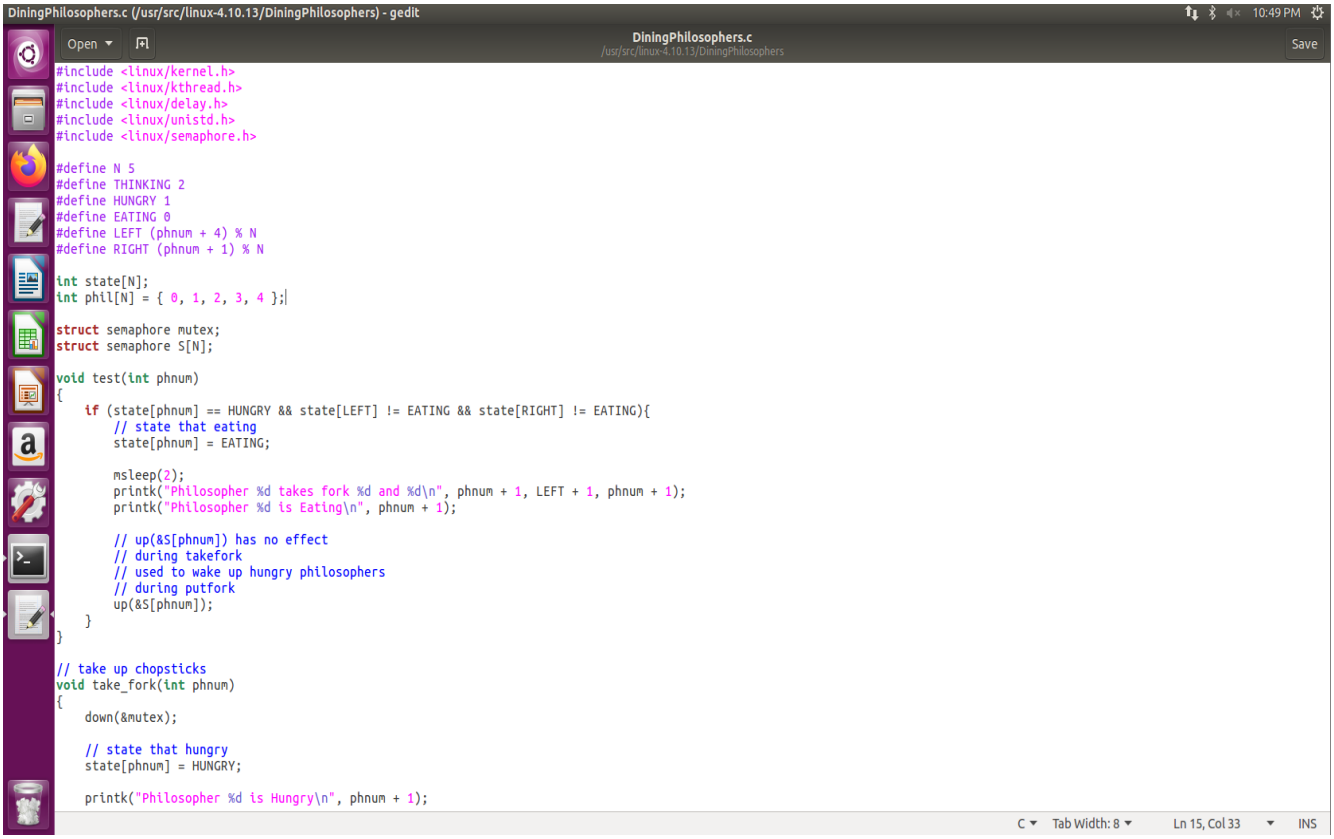
Image-Descriptions:

Image-1 to Image-4: Dining Philosophers' Code

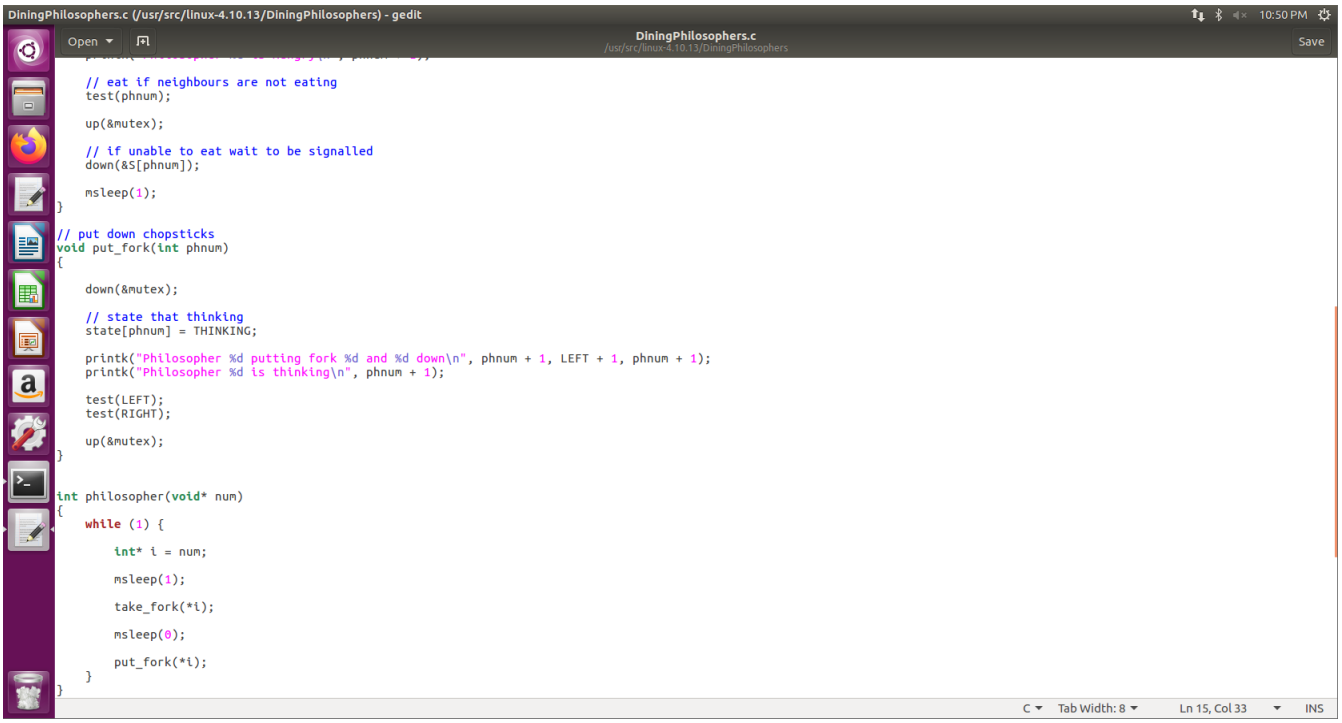
Image-5 to Image-8: Dining Philosophers' System Call's results

1.

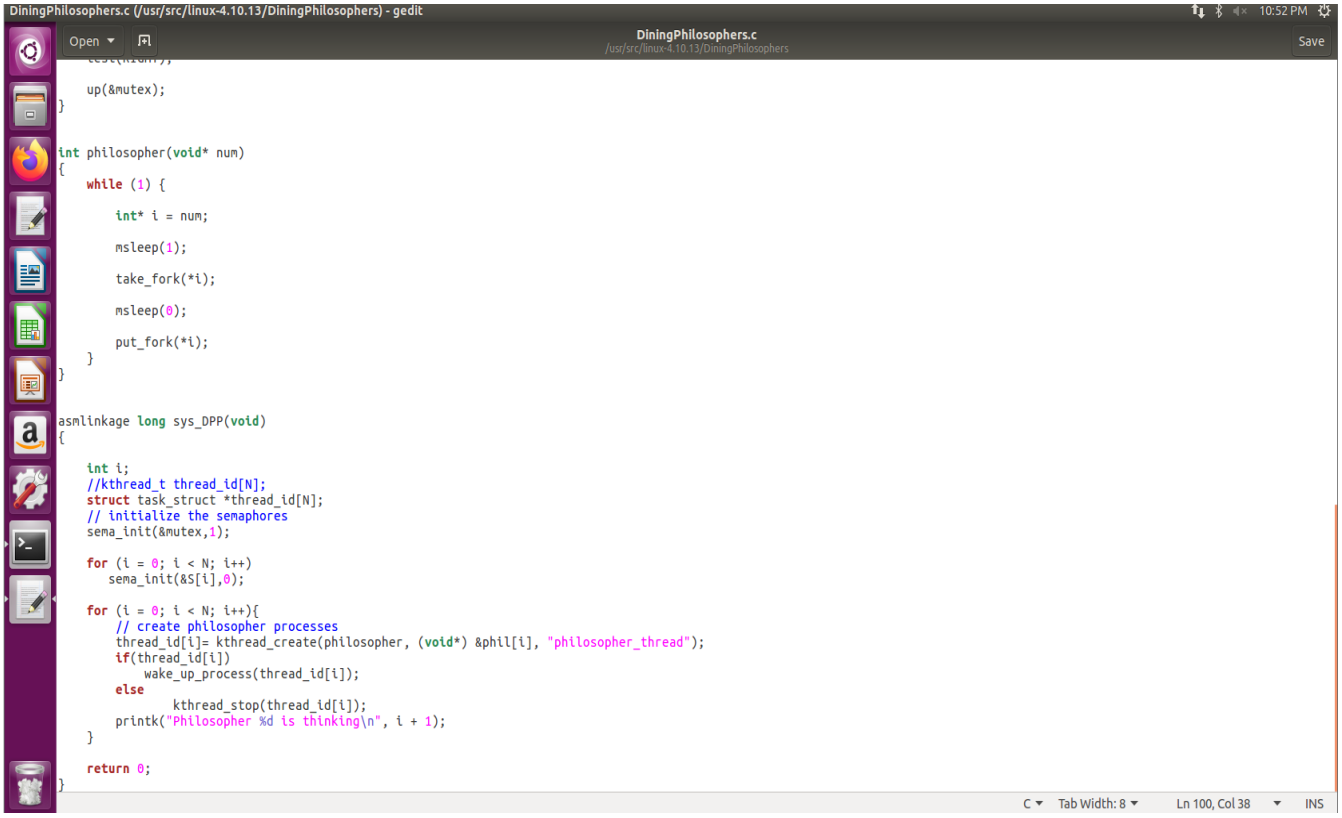
2.



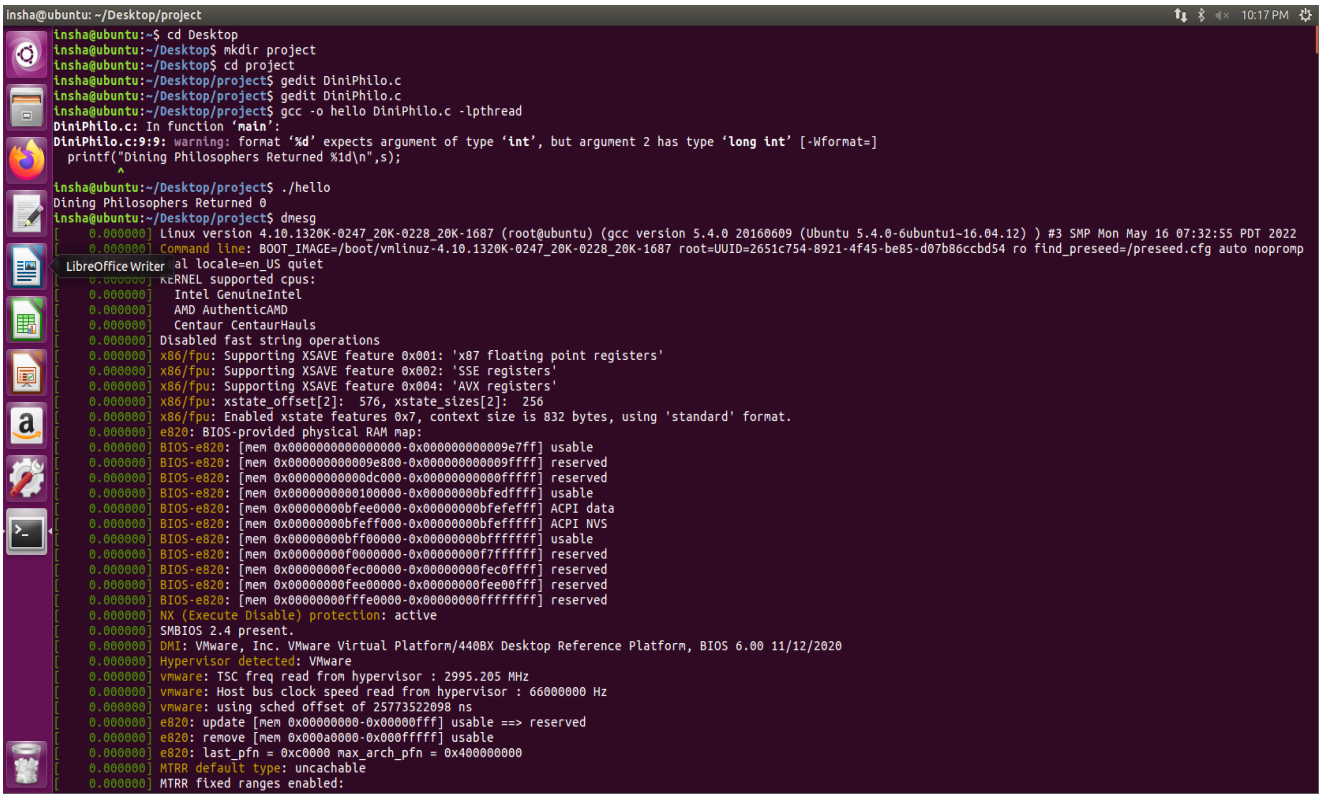
3.


DiningPhilosophers.c (/usr/src/linux-4.10.13/DiningPhilosophers) - gedit
10:50 PM
Open
DiningPhilosophers.c
/usr/src/linux-4.10.13/DiningPhilosophers
Save
// eat if neighbours are not eating
test(phnum);
up(&mutex);
// if unable to eat wait to be signalled
down(&S[phnum]);
msleep(1);
}
// put down chopsticks
void put_fork(int phnum)
{
down(&mutex);
// state that thinking
state[phnum] = THINKING;
printk("Philosopher %d putting fork %d and %d down\n", phnum + 1, LEFT + 1, phnum + 1);
printk("Philosopher %d is thinking\n", phnum + 1);
test(LEFT);
test(RIGHT);
up(&mutex);
}
int philosopher(void* num)
{
while (1) {
int* i = num;
msleep(1);
take_fork(*i);
msleep(0);
put_fork(*i);
}
}
C
Tab Width: 8
Ln 15, Col 33
INS

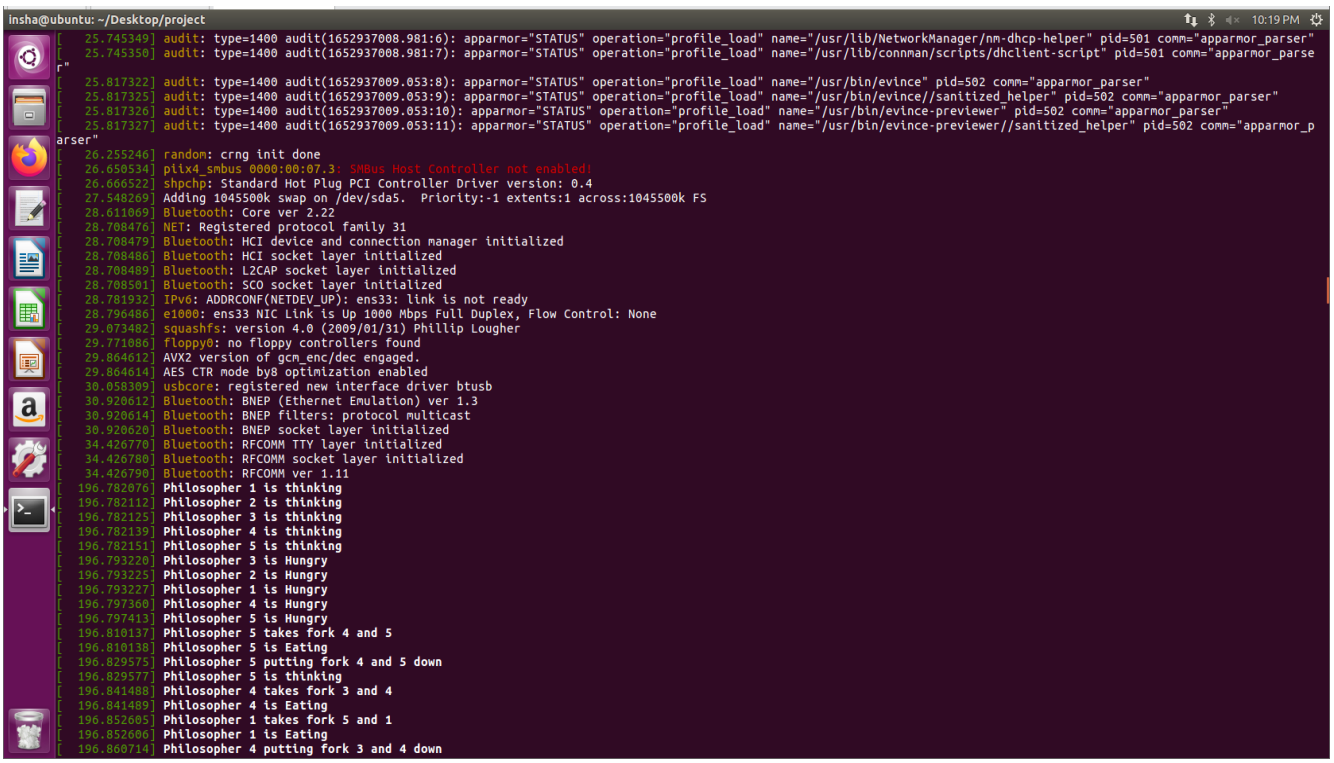
4.


DiningPhilosophers.c (/usr/src/linux-4.10.13/DiningPhilosophers) - gedit
10:52 PM
Open
DiningPhilosophers.c
/usr/src/linux-4.10.13/DiningPhilosophers
Save
up(&mutex);
}
int philosopher(void* num)
{
while (1) {
int* i = num;
msleep(1);
take_fork(*i);
msleep(0);
put_fork(*i);
}
}
asmlinkage long sys_DPP(void)
{
int i;
//kthread_t thread_id[N];
struct task_struct *thread_id[N];
// initialize the semaphores
sema_init(&mutex,1);
for (i = 0; i < N; i++)
sema_init(&S[i],0);
for (i = 0; i < N; i++){
// create philosopher processes
thread_id[i]= kthread_create(philosopher, (void*) &phil[i], "philosopher_thread");
if(thread_id[i])
wake_up_process(thread_id[i]);
else
kthread_stop(thread_id[i]);
printk("Philosopher %d is thinking\n", i + 1);
}
return 0;
}
C
Tab Width: 8
Ln 100, Col 38
INS

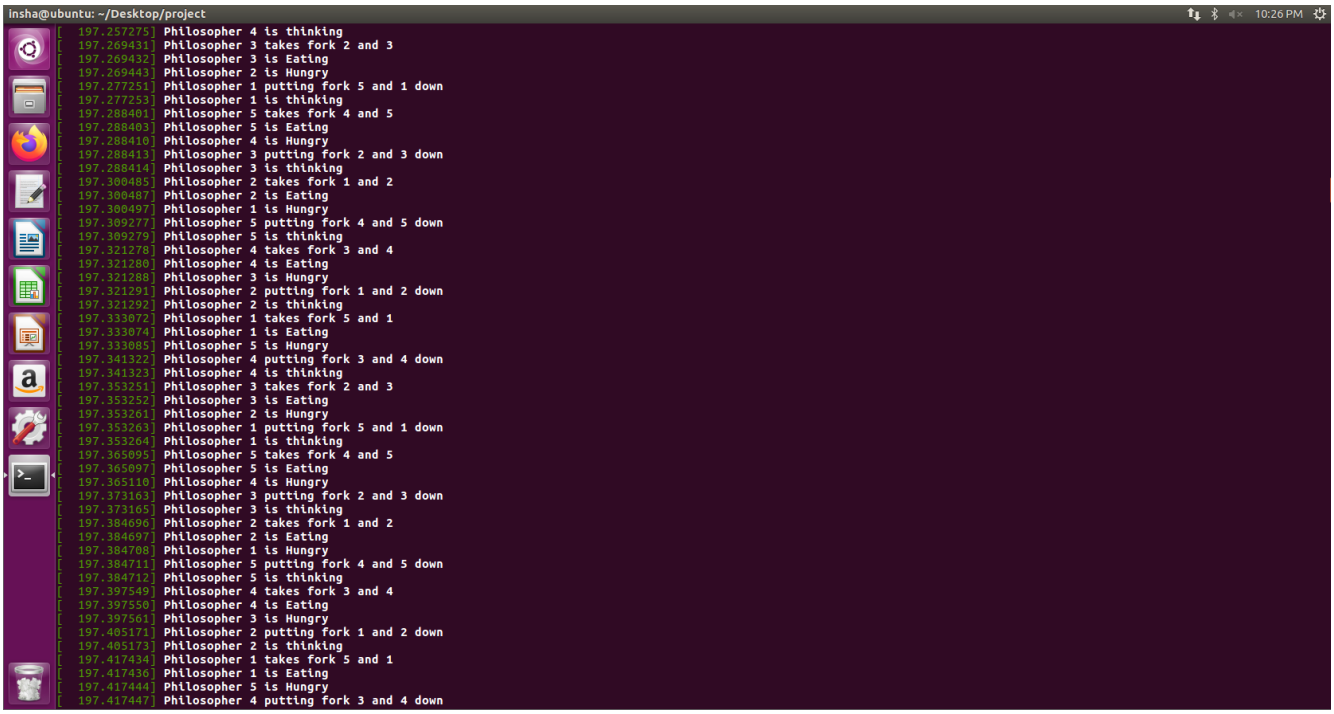
5.


insha@ubuntu: ~/Desktop/project
10:17 PM
insha@ubuntu:~$ cd Desktop
insha@ubuntu:~/Desktop$ mkdir project
insha@ubuntu:~/Desktop$ cd project
insha@ubuntu:~/Desktop/project$ gedit DiniPhilo.c
insha@ubuntu:~/Desktop/project$ gedit DiniPhilo.c
insha@ubuntu:~/Desktop/project$ gcc -o hello DiniPhilo.c -lpthread
DiniPhilo.c: In function 'main':
DiniPhilo.c:9:9: warning: format '%d' expects argument of type 'int', but argument 2 has type 'long int' [-Wformat=]
printf("Dining Philosophers Returned %id\n",s);
^
insha@ubuntu:~/Desktop/project$ ./hello
Dining Philosophers Returned 0
insha@ubuntu:~/Desktop/project$ dmesg
[0.000000] Linux version 4.10.1320K-0247_20K-0228_20K-1687 (root@ubuntu) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.12)) #3 SMP Mon May 16 07:32:55 PDT 2022
[0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.10.1320K-0247_20K-0228_20K-1687 root=UUID=2651c754-8921-4f45-be85-d07b86ccbd54 ro find_preseed=/preseed.cfg auto nopromp
LibreOffice Writer
al locale=en_US quiet
KERNEL supported cpus:
[0.000000] Intel GenuineIntel
[0.000000] AMD AuthenticAMD
[0.000000] Centaur CentaurHauls
[0.000000] Disabled fast string operations
[0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[0.000000] e820: BIOS-provided physical RAM map:
[0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009e7ff] usable
[0.000000] BIOS-e820: [mem 0x000000000009e800-0x000000000009ffff] reserved
[0.000000] BIOS-e820: [mem 0x00000000000dc000-0x00000000000fffff] reserved
[0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000bfedffff] usable
[0.000000] BIOS-e820: [mem 0x00000000bfee0000-0x00000000bfefefff] ACPI data
[0.000000] BIOS-e820: [mem 0x00000000bfeff000-0x00000000bfefffff] ACPI NVS
[0.000000] BIOS-e820: [mem 0x00000000bff00000-0x00000000bfffffff] usable
[0.000000] BIOS-e820: [mem 0x00000000f0000000-0x00000000f7ffffff] reserved
[0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec0ffff] reserved
[0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[0.000000] BIOS-e820: [mem 0x00000000fffe0000-0x00000000ffffffff] reserved
[0.000000] NX (Execute Disable) protection: active
[0.000000] SMBIOS 2.4 present.
[0.000000] DMI: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020
[0.000000] Hypervisor detected: VMware
[0.000000] vmware: TSC freq read from hypervisor : 2995.205 MHz
[0.000000] vmware: Host bus clock speed read from hypervisor : 66000000 Hz
[0.000000] vmware: using sched offset of 25773522098 ns
[0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[0.000000] e820: last_pfn = 0xc0000 max_arch_pfn = 0x400000000
[0.000000] MTRR default type: uncachable
[0.000000] MTRR fixed ranges enabled:

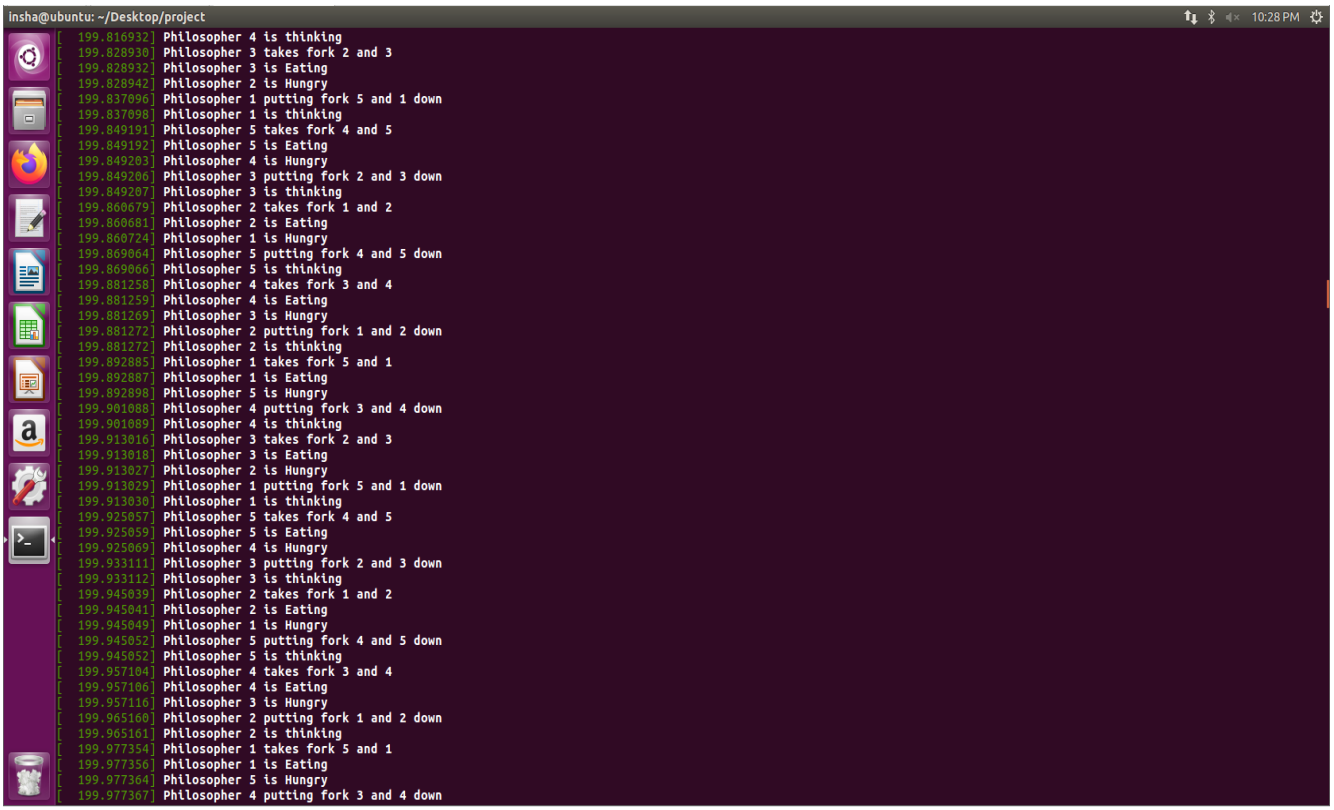
6.


insha@ubuntu: ~/Desktop/project
10:19 PM
[25.745349] audit: type=1400 audit(1652937008.981:6): apparmor="STATUS" operation="profile_load" name="/usr/lib/NetworkManager/nm-dhcp-helper" pid=501 comm="apparmor_parser"
[25.745350] audit: type=1400 audit(1652937008.981:7): apparmor="STATUS" operation="profile_load" name="/usr/lib/connman/scripts/dhclient-script" pid=501 comm="apparmor_parse
r"
[25.817322] audit: type=1400 audit(1652937009.053:8): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince" pid=502 comm="apparmor_parser"
[25.817325] audit: type=1400 audit(1652937009.053:9): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince//sanitized_helper" pid=502 comm="apparmor_parser"
[25.817326] audit: type=1400 audit(1652937009.053:10): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince-previewer" pid=502 comm="apparmor_parser"
[25.817327] audit: type=1400 audit(1652937009.053:11): apparmor="STATUS" operation="profile_load" name="/usr/bin/evince-previewer//sanitized_helper" pid=502 comm="apparmor_p
arser"
[26.255246] random: crng init done
[26.650534] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
[26.666522] shpchp: Standard Hot Plug PCI Controller Driver version: 0.4
[27.548269] Adding 1045500k swap on /dev/sda5. Priority:-1 extents:1 across:1045500k FS
[28.611069] Bluetooth: Core ver 2.22
[28.708476] NET: Registered protocol family 31
[28.708479] Bluetooth: HCI device and connection manager initialized
[28.708486] Bluetooth: HCI socket layer initialized
[28.708489] Bluetooth: L2CAP socket layer initialized
[28.708501] Bluetooth: SCO socket layer initialized
[28.781932] IPv6: ADDRCONF(NETDEV_UP): ens33: link is not ready
[28.796486] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[29.073482] squashfs: version 4.0 (2009/01/31) Phillip Lougher
[29.771086] floppy0: no floppy controllers found
[29.864612] AVX2 version of gcm_enc/dec engaged.
[29.864614] AES CTR mode by8 optimization enabled
[30.058309] usbcore: registered new interface driver btusb
[30.920612] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[30.920614] Bluetooth: BNEP filters: protocol multicast
[30.920620] Bluetooth: BNEP socket layer initialized
[34.426770] Bluetooth: RFCOMM TTY layer initialized
[34.426780] Bluetooth: RFCOMM socket layer initialized
[34.426790] Bluetooth: RFCOMM ver 1.11
[196.782076] Philosopher 1 is thinking
[196.782112] Philosopher 2 is thinking
[196.782125] Philosopher 3 is thinking
[196.782139] Philosopher 4 is thinking
[196.782151] Philosopher 5 is thinking
[196.793220] Philosopher 3 is Hungry
[196.793225] Philosopher 2 is Hungry
[196.793227] Philosopher 1 is Hungry
[196.797360] Philosopher 4 is Hungry
[196.797413] Philosopher 5 is Hungry
[196.810137] Philosopher 5 takes fork 4 and 5
[196.810138] Philosopher 5 is Eating
[196.829575] Philosopher 5 putting fork 4 and 5 down
[196.829577] Philosopher 5 is thinking
[196.841488] Philosopher 4 takes fork 3 and 4
[196.841489] Philosopher 4 is Eating
[196.852605] Philosopher 1 takes fork 5 and 1
[196.852606] Philosopher 1 is Eating
[196.860714] Philosopher 4 putting fork 3 and 4 down

7.


insha@ubuntu: ~/Desktop/project
10:26 PM
[197.257275] Philosopher 4 is thinking
[197.269431] Philosopher 3 takes fork 2 and 3
[197.269432] Philosopher 3 is Eating
[197.269443] Philosopher 2 is Hungry
[197.277251] Philosopher 1 putting fork 5 and 1 down
[197.277253] Philosopher 1 is thinking
[197.288401] Philosopher 5 takes fork 4 and 5
[197.288403] Philosopher 5 is Eating
[197.288410] Philosopher 4 is Hungry
[197.288413] Philosopher 3 putting fork 2 and 3 down
[197.288414] Philosopher 3 is thinking
[197.300485] Philosopher 2 takes fork 1 and 2
[197.300487] Philosopher 2 is Eating
[197.300497] Philosopher 1 is Hungry
[197.309277] Philosopher 5 putting fork 4 and 5 down
[197.309279] Philosopher 5 is thinking
[197.321278] Philosopher 4 takes fork 3 and 4
[197.321280] Philosopher 4 is Eating
[197.321288] Philosopher 3 is Hungry
[197.321291] Philosopher 2 putting fork 1 and 2 down
[197.321292] Philosopher 2 is thinking
[197.333072] Philosopher 1 takes fork 5 and 1
[197.333074] Philosopher 1 is Eating
[197.333085] Philosopher 5 is Hungry
[197.341322] Philosopher 4 putting fork 3 and 4 down
[197.341323] Philosopher 4 is thinking
[197.353251] Philosopher 3 takes fork 2 and 3
[197.353252] Philosopher 3 is Eating
[197.353261] Philosopher 2 is Hungry
[197.353263] Philosopher 1 putting fork 5 and 1 down
[197.353264] Philosopher 1 is thinking
[197.365095] Philosopher 5 takes fork 4 and 5
[197.365097] Philosopher 5 is Eating
[197.365110] Philosopher 4 is Hungry
[197.373163] Philosopher 3 putting fork 2 and 3 down
[197.373165] Philosopher 3 is thinking
[197.384696] Philosopher 2 takes fork 1 and 2
[197.384697] Philosopher 2 is Eating
[197.384708] Philosopher 1 is Hungry
[197.384711] Philosopher 5 putting fork 4 and 5 down
[197.384712] Philosopher 5 is thinking
[197.397549] Philosopher 4 takes fork 3 and 4
[197.397550] Philosopher 4 is Eating
[197.397561] Philosopher 3 is Hungry
[197.405171] Philosopher 2 putting fork 1 and 2 down
[197.405173] Philosopher 2 is thinking
[197.417434] Philosopher 1 takes fork 5 and 1
[197.417436] Philosopher 1 is Eating
[197.417444] Philosopher 5 is Hungry
[197.417447] Philosopher 4 putting fork 3 and 4 down

8.


insha@ubuntu: ~/Desktop/project
10:28 PM
[199.816932] Philosopher 4 is thinking
[199.828930] Philosopher 3 takes fork 2 and 3
[199.828932] Philosopher 3 is Eating
[199.828942] Philosopher 2 is Hungry
[199.837096] Philosopher 1 putting fork 5 and 1 down
[199.837098] Philosopher 1 is thinking
[199.849191] Philosopher 5 takes fork 4 and 5
[199.849192] Philosopher 5 is Eating
[199.849203] Philosopher 4 is Hungry
[199.849206] Philosopher 3 putting fork 2 and 3 down
[199.849207] Philosopher 3 is thinking
[199.860679] Philosopher 2 takes fork 1 and 2
[199.860681] Philosopher 2 is Eating
[199.860724] Philosopher 1 is Hungry
[199.869064] Philosopher 5 putting fork 4 and 5 down
[199.869066] Philosopher 5 is thinking
[199.881258] Philosopher 4 takes fork 3 and 4
[199.881259] Philosopher 4 is Eating
[199.881269] Philosopher 3 is Hungry
[199.881272] Philosopher 2 putting fork 1 and 2 down
[199.881272] Philosopher 2 is thinking
[199.892885] Philosopher 1 takes fork 5 and 1
[199.892887] Philosopher 1 is Eating
[199.892898] Philosopher 5 is Hungry
[199.901088] Philosopher 4 putting fork 3 and 4 down
[199.901089] Philosopher 4 is thinking
[199.913016] Philosopher 3 takes fork 2 and 3
[199.913018] Philosopher 3 is Eating
[199.913027] Philosopher 2 is Hungry
[199.913029] Philosopher 1 putting fork 5 and 1 down
[199.913030] Philosopher 1 is thinking
[199.925057] Philosopher 5 takes fork 4 and 5
[199.925059] Philosopher 5 is Eating
[199.925069] Philosopher 4 is Hungry
[199.933111] Philosopher 3 putting fork 2 and 3 down
[199.933112] Philosopher 3 is thinking
[199.945039] Philosopher 2 takes fork 1 and 2
[199.945041] Philosopher 2 is Eating
[199.945049] Philosopher 1 is Hungry
[199.945052] Philosopher 5 putting fork 4 and 5 down
[199.945052] Philosopher 5 is thinking
[199.957104] Philosopher 4 takes fork 3 and 4
[199.957106] Philosopher 4 is Eating
[199.957116] Philosopher 3 is Hungry
[199.965160] Philosopher 2 putting fork 1 and 2 down
[199.965161] Philosopher 2 is thinking
[199.977354] Philosopher 1 takes fork 5 and 1
[199.977356] Philosopher 1 is Eating
[199.977364] Philosopher 5 is Hungry
[199.977367] Philosopher 4 putting fork 3 and 4 down

CONCLUSION:

To ensure each philosopher gets an opportunity to eat, we'll be going clockwise around the table each philosopher picks up the fork to his right. If it is unavailable, the code blocks the philosopher to take the right fork and puts in the ***waiting queue***. Iterate again going clockwise and each philosopher who has access to his left & right fork starts eating. It will ***prevent deadlocks*** in each iteration (where N being the number of philosophers, that is equal to 5 in our case). Moreover, to initialize the philosophers eating we have assigned an identity to each philosopher.

References:

https://phoenixnap.com/kb/build-linux-kernel

https://www.youtube.com/watch?v=AP-tBd84vbM

https://www.ee.ryerson.ca/~courses/coe518/Labs/lab4/lisi.edu-dining-Philosopherecture8.pdf

https://www.geeksforgeeks.org/dining-philosopher-problem-using-semaphores/