**EXERCISES**

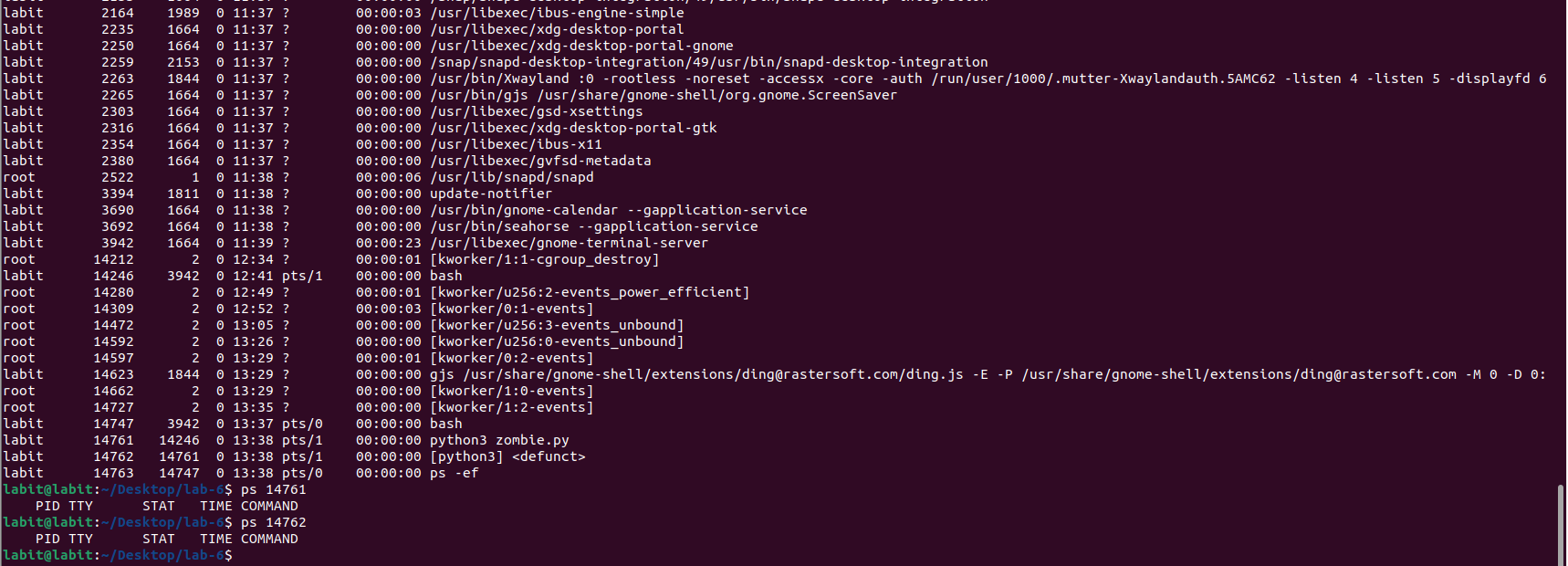
1. Using a Linux system, write a program that forks a child process that ultimately becomes a zombie

process. This zombie process must remain in the system for at least 10 seconds.

Ans:

Zombie.py

|  |
| --- |
| import os  import time  pid = os.fork()  if pid > 0:  print(f"Parent process started. PID: {os.getpid()}")  print(f"Child process PID: {pid}")  print("Parent sleeping for 10 seconds to let the child become a zombie...")  time.sleep(10) # Parent sleeps for 10 seconds to let the child become a zombie  print("Parent exiting.")  elif pid == 0:  print(f"Child process created. PID: {os.getpid()}")  os.\_exit(0) # Child process exits immediately, creating a zombie process |

OUTPUT:

2. Write a program that creates a child process which further creates its two child processes. Display the

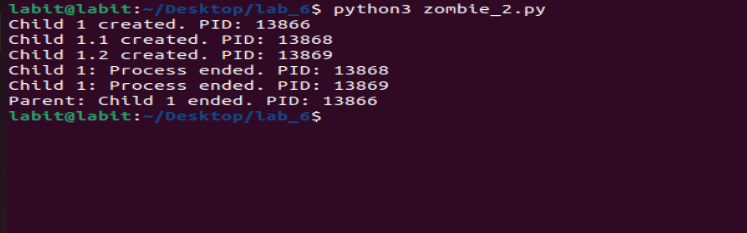
process id of the terminated child to understand the hierarchy of termination of each child process.

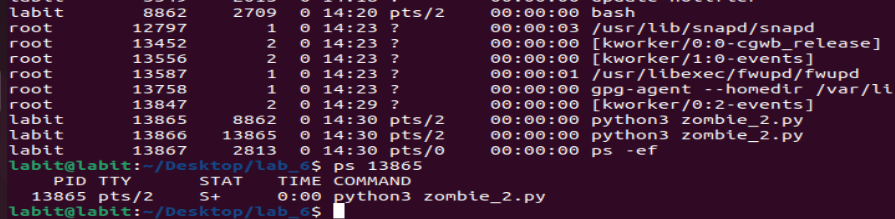
Ans:

Zombie\_2.py:

|  |
| --- |
| import os  import time  pid = os.fork()  if pid == 0:  # This is Child 1  print("Child 1 created. PID:", os.getpid())  time.sleep(10) # Delay so you can see this process in ps  pid1 = os.fork()  if pid1 == 0:  print("Child 1.1 created. PID:", os.getpid())  time.sleep(10)  os.\_exit(0)  pid2 = os.fork()  if pid2 == 0:  print("Child 1.2 created. PID:", os.getpid())  time.sleep(10)  os.\_exit(0)  for \_ in range(2):  ended\_pid, \_ = os.wait()  print("Child 1: Process ended. PID:", ended\_pid)  os.\_exit(0)  else:  ended\_pid, \_ = os.wait()  print("Parent: Child 1 ended. PID:", ended\_pid) |

OUTPUT:





3. Write a program that create child process then parent call the wait() function to know status of child

also display process id of both processes.

Ans:

Zombie\_3.py:

|  |
| --- |
| import os  import time  def main():  pid = os.fork()  if pid == 0:  print(f"Child Process: PID = {os.getpid()}, Parent PID = {os.getppid()}")  time.sleep(60)  os.\_exit(0)  else:  print(f"Parent Process: PID = {os.getpid()} waiting for child...")  os.wait()  print("Child process terminated. Parent resumes.")  if \_\_name\_\_ == "\_\_main\_\_":  main() |

OUTPUT:

