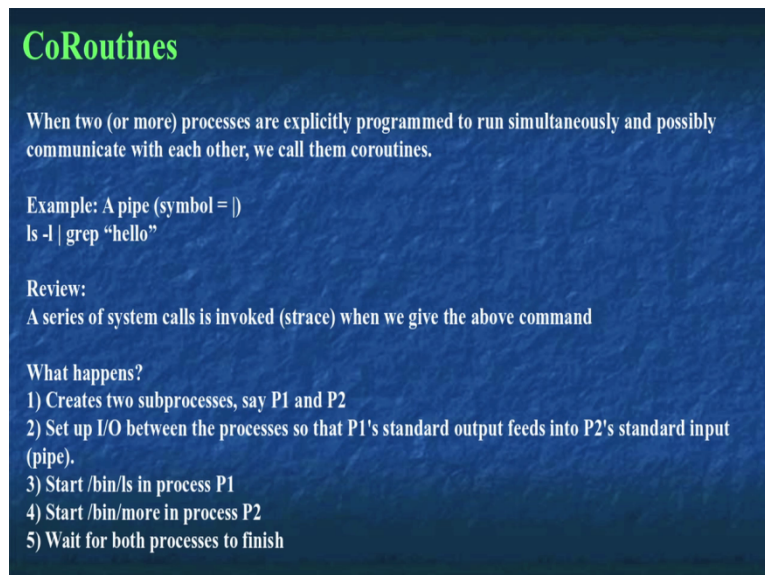


Information Security 3
Sri M J Shankar Raman,
Consultant Department of Computer Science and Engineering,
Indian Institute of Technology Madras
Module 39
Shell Coroutines

Hello there, welcome to this session on co routines. What we will see in this session is how we can run two programs simultaneously in a shell as well as what happens if I run these two programs through a script a shell script. So before we go into how to run programs in the background and how to run it using shell scripts.

(Refer Slide Time: 00:48)



CoRoutines

When two (or more) processes are explicitly programmed to run simultaneously and possibly communicate with each other, we call them coroutines.

Example: A pipe (symbol = |)
`ls -l | grep "hello"`

Review:
A series of system calls is invoked (strace) when we give the above command

What happens?

- 1) Creates two subprocesses, say P1 and P2
- 2) Set up I/O between the processes so that P1's standard output feeds into P2's standard input (pipe).
- 3) Start /bin/ls in process P1
- 4) Start /bin/more in process P2
- 5) Wait for both processes to finish

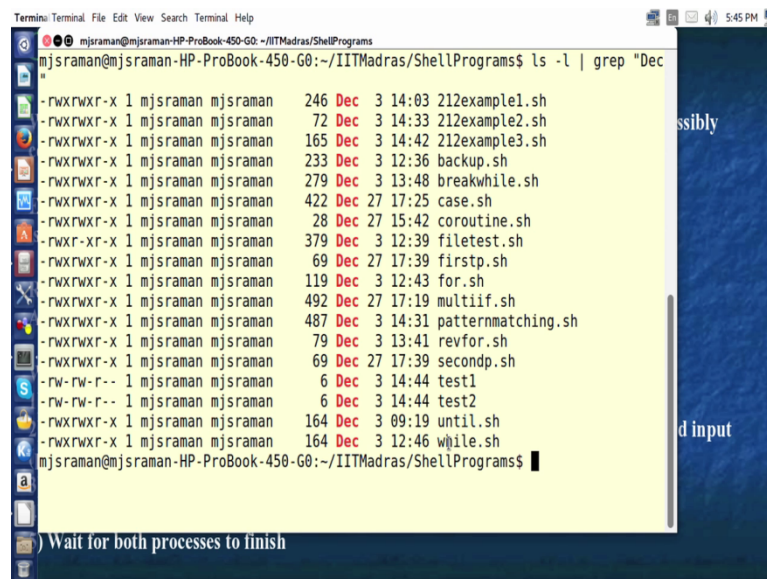
Let us try to understand what is a Coroutine. When two or more processes are explicitly programmed to run simultaneously possibly they are communicating with each other then we call them as co routines. So let us take an example.

We know that if we type a command called `ls space minus l` and then we type it to `grep` what will happen is that it creates two processes the first process is `ls space minus l` which is actually a series of system calls and then `grep` is another process which is another series of system calls and then these two processes are related through a pipe symbol which essentially means that the output of process 1 becomes the input of process two, that is the standard output of process 1 becomes the standard input of process 2.

And here there are two things that are happening one the programs are running in parallel and two one program is communicating with the other program. Now what happens the shell what it does is that it waits for both the processes to finish and then it comes out and then user will be able to type in more commands.

So in this case for example let us try this example and see what happens.

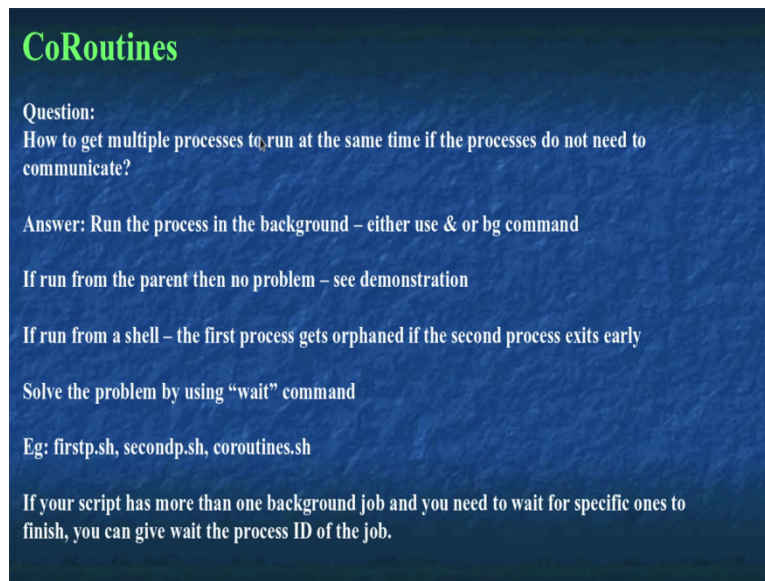
(Refer Slide Time: 02:30)



```
Terminal: Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ls -l | grep "Dec"
-rwxrwxr-x 1 mjsraman mjsraman 246 Dec 3 14:03 212example1.sh
-rwxrwxr-x 1 mjsraman mjsraman 72 Dec 3 14:33 212example2.sh
-rwxrwxr-x 1 mjsraman mjsraman 165 Dec 3 14:42 212example3.sh
-rwxrwxr-x 1 mjsraman mjsraman 233 Dec 3 12:36 backup.sh
-rwxrwxr-x 1 mjsraman mjsraman 279 Dec 3 13:48 breakwhile.sh
-rwxrwxr-x 1 mjsraman mjsraman 422 Dec 27 17:25 case.sh
-rwxrwxr-x 1 mjsraman mjsraman 28 Dec 27 15:42 coroutine.sh
-rwxr-xr-x 1 mjsraman mjsraman 379 Dec 3 12:39 filetest.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:39 firstp.sh
-rwxrwxr-x 1 mjsraman mjsraman 119 Dec 3 12:43 for.sh
-rwxrwxr-x 1 mjsraman mjsraman 492 Dec 27 17:19 multiif.sh
-rwxrwxr-x 1 mjsraman mjsraman 487 Dec 3 14:31 patternmatching.sh
-rwxrwxr-x 1 mjsraman mjsraman 79 Dec 3 13:41 revfor.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:39 secondp.sh
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test1
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test2
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 09:19 until.sh
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 12:46 while.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
```

So here what I do is I put ls minus l and then I put a grep and then I say let us say I put December I want to search for December. So what is happening is that it first after both the programs are executed that is ls minus l and grep are executed it returns back to the shell. Now essentially what happens is that the shell waits for both the processes to finish and then it comes out. Now let us consider another scenario.

(Refer Slide Time: 03:06)



CoRoutines

Question:
How to get multiple processes to run at the same time if the processes do not need to communicate?

Answer: Run the process in the background – either use & or bg command

If run from the parent then no problem – see demonstration

If run from a shell – the first process gets orphaned if the second process exits early

Solve the problem by using “wait” command

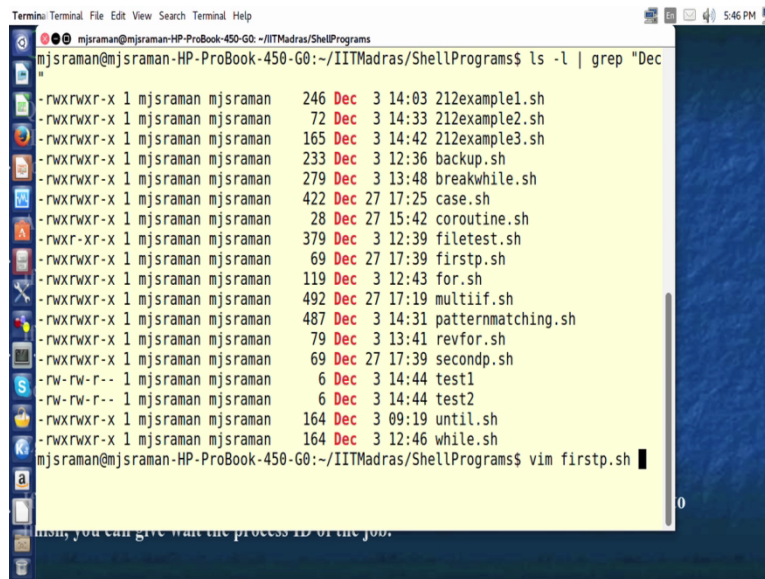
Eg: firstp.sh, secondp.sh, coroutines.sh

If your script has more than one background job and you need to wait for specific ones to finish, you can give wait the process ID of the job.

So in this scenario what we are going to do is that we are going to find out how to get multiple processes to run at the same time if the processes do not need to communicate with each other so remember in the previous case what happened is that I gave ls minus l command and the output of the ls minus l became the input of the grep command. Now let us say that we do not want such communication to happen that means each process must execute independently of each other.

So for that what we can do is we can run the processes in the background and for this either you can use the and symbol or you can use the bg command I mean there are two ways what you do is you just run the program then to make the process go to sleep and then you type the bg command and then currently executing process actually goes to the background and then you can start typing new commands.

(Refer Slide Time: 04:00)



A terminal window showing the output of the command `ls -l | grep "Dec"`. The output lists various shell scripts with their permissions, owner, group, size, date, and filename. The scripts include `212example1.sh`, `212example2.sh`, `212example3.sh`, `backup.sh`, `breakwhile.sh`, `case.sh`, `coroutine.sh`, `filetest.sh`, `firstp.sh`, `for.sh`, `multiif.sh`, `patternmatching.sh`, `revfor.sh`, `secondp.sh`, `test1`, `test2`, `until.sh`, and `while.sh`. The terminal also shows the command `vim firstp.sh` being entered.

```
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ls -l | grep "Dec"
-rwxrwxr-x 1 mjsraman mjsraman 246 Dec 3 14:03 212example1.sh
-rwxrwxr-x 1 mjsraman mjsraman 72 Dec 3 14:33 212example2.sh
-rwxrwxr-x 1 mjsraman mjsraman 165 Dec 3 14:42 212example3.sh
-rwxrwxr-x 1 mjsraman mjsraman 233 Dec 3 12:36 backup.sh
-rwxrwxr-x 1 mjsraman mjsraman 279 Dec 3 13:48 breakwhile.sh
-rwxrwxr-x 1 mjsraman mjsraman 422 Dec 27 17:25 case.sh
-rwxrwxr-x 1 mjsraman mjsraman 28 Dec 27 15:42 coroutine.sh
-rwxr-xr-x 1 mjsraman mjsraman 379 Dec 3 12:39 filetest.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:39 firstp.sh
-rwxrwxr-x 1 mjsraman mjsraman 119 Dec 3 12:43 for.sh
-rwxrwxr-x 1 mjsraman mjsraman 492 Dec 27 17:19 multiif.sh
-rwxrwxr-x 1 mjsraman mjsraman 487 Dec 3 14:31 patternmatching.sh
-rwxrwxr-x 1 mjsraman mjsraman 79 Dec 3 13:41 revfor.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:39 secondp.sh
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test1
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test2
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 09:19 until.sh
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 12:46 while.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim firstp.sh
```

Now the question is so let us try to see how to use the background process. So I have two programs, so let us say I call them as firstp dot sh.

(Refer Slide Time: 04:13)

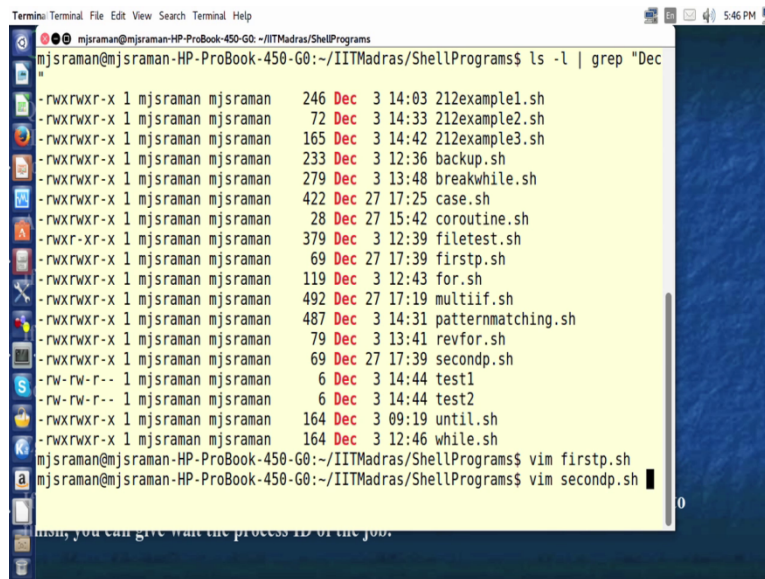


A terminal window showing the contents of a shell script named `firstp.sh`. The script initializes a variable `n` to 10 and enters a `while` loop that decrements `n` by 1, prints the value, and sleeps for 5 seconds. The terminal also shows the command `vim firstp.sh` being entered.

```
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
1 n=10
2
3 while [ $n -gt 1 ]
4 do
5     n=`expr $n - 1`
6     sleep 5
7     echo $n
8 done
9
```

So what this program simply does is it just takes initializes value of n to 10 then it goes on decrementing 10 by 1 so it prints 9,8,7,6,5... and then between printing 9,8,7,6,5... and so on it sleeps for about 5 seconds.

(Refer Slide Time: 04:30)



```
Terminal: Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ls -l | grep "Dec"
-rwxrwxr-x 1 mjsraman mjsraman 246 Dec 3 14:03 212example1.sh
-rwxrwxr-x 1 mjsraman mjsraman 72 Dec 3 14:33 212example2.sh
-rwxrwxr-x 1 mjsraman mjsraman 165 Dec 3 14:42 212example3.sh
-rwxrwxr-x 1 mjsraman mjsraman 233 Dec 3 12:36 backup.sh
-rwxrwxr-x 1 mjsraman mjsraman 279 Dec 3 13:48 breakwhile.sh
-rwxrwxr-x 1 mjsraman mjsraman 422 Dec 27 17:25 case.sh
-rwxrwxr-x 1 mjsraman mjsraman 28 Dec 27 15:42 coroutine.sh
-rwxr-xr-x 1 mjsraman mjsraman 379 Dec 3 12:39 filetest.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:39 firstp.sh
-rwxrwxr-x 1 mjsraman mjsraman 119 Dec 3 12:43 for.sh
-rwxrwxr-x 1 mjsraman mjsraman 492 Dec 27 17:19 multiif.sh
-rwxrwxr-x 1 mjsraman mjsraman 487 Dec 3 14:31 patternmatching.sh
-rwxrwxr-x 1 mjsraman mjsraman 79 Dec 3 13:41 revfor.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:39 secondp.sh
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test1
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test2
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 09:19 until.sh
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 12:46 while.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim firstp.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim secondp.sh
```

Let me take another program which is known as second program dot sh.

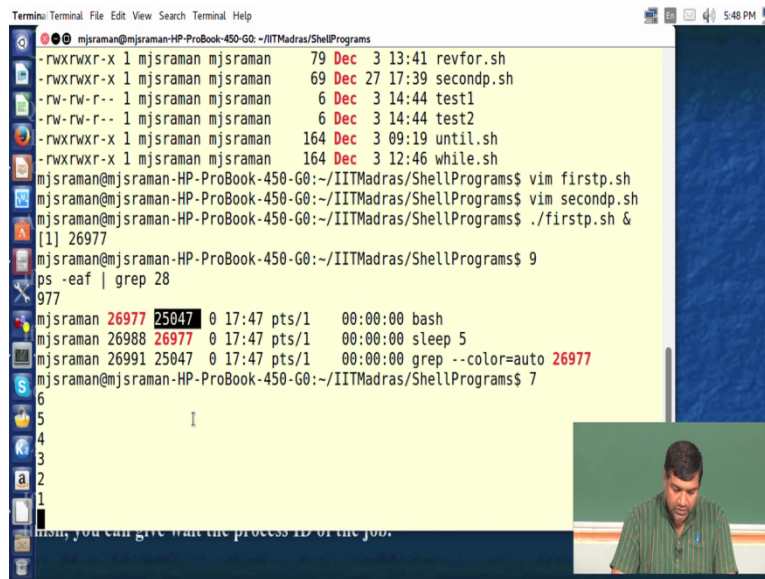
(Refer Slide Time: 04:36)



```
Terminal: Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
1 n=10
2
3 while [ $n -gt 1 ]
4 do
5     n=`expr $n - 1`
6     sleep 1
7     echo $n
8 done
9
"secondp.sh" 9L, 69C
6,1-4 All
```

So in this program it is the same logic except that i sleep for only 1 second so it essentially says start these two programs then if I start at the appropriate time the second program should end before the first program and if you run the first program in the background and the first program in the foreground what will happen is that you will get the shell prompt as soon as the second program exits.

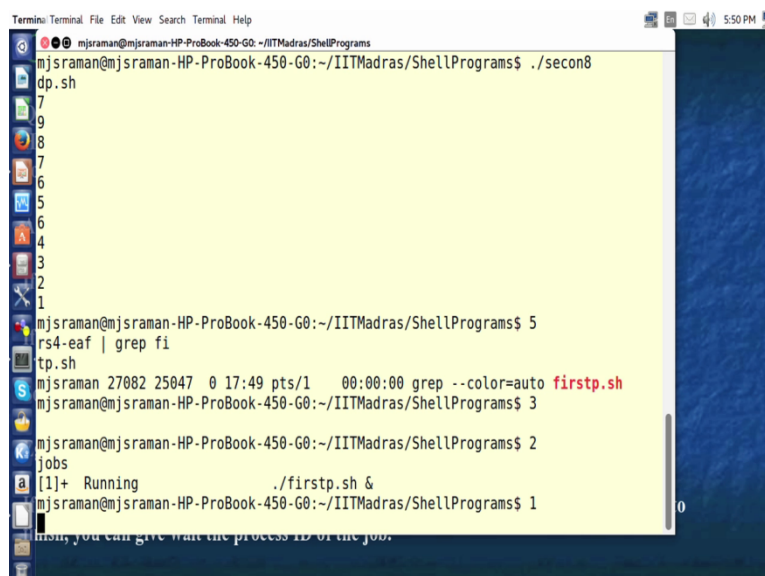
(Refer Slide Time: 05:00)



```
Termin@Terminal: File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
-rwxrwxr-x 1 mjsraman mjsraman 79 Dec 3 13:41 revfor.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:39 secondp.sh
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test1
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test2
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 09:19 until.sh
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 12:46 while.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim firstp.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim secondp.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./firstp.sh &
[1] 26977
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 9
ps -eaf | grep 28
977
mjsraman 26977 25047 0 17:47 pts/1 00:00:00 bash
mjsraman 26988 26977 0 17:47 pts/1 00:00:00 sleep 5
mjsraman 26991 25047 0 17:47 pts/1 00:00:00 grep --color=auto 26977
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 7
6
5
4
3
2
1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
```

So let us try to see what is happening here, so if you look at this I can just say type firstp dot sh and then I move in to the background so this program will start working the process ID is 26977 and then I can just see what happens with this process ID 26977. So it says 26977 the most important thing to look at is that the process ID 26977 the parent process is 25047 ok 25047 which is the bash. So parent of 26977 is 25047 so keep this in mind this is very important because this is where we are going to explain so something so now again coming back so let us now try to let this program end so that we will now run both the programs in the background so both the first program and the second program.

(Refer Slide Time: 06:00)



```
Termin@Terminal: File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./secondp.sh
dp.sh
7
9
8
7
6
5
6
4
3
2
1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 5
rs4-eaf | grep fi
tp.sh
mjsraman 27082 25047 0 17:49 pts/1 00:00:00 grep --color=auto firstp.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 3
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 2
jobs
[1]+  Running                  ./firstp.sh &
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
```

So this program is ended so we will run the both the programs in the background so first dot sh is run then I will also run second dot sh. So what happens is that the second program dot sh will actually finish before so it starts 9,8,7 so it goes very fast so it finishes before the first program. But then you start getting the shell prompt so now I can take a look at the shell prompt and then (06:30)for first dot sh so then the first program also exits and so on so what we can do is so the first program is executing so if you see the first program is running so idea is that these two programs are working independent of each other. So this is an example of a Coroutines.

Now let us go back and if you look at what had happened see if you look at this if you look at this program 26977 ok and the bash shell was here so the parent shell was here ok. Now and your parent shell was live when I say live when I say the parent shell was live you had a program that was working on a foreground you had a program that was working on the background and both came out of the same shell.

Now let us consider a case where we put these two programs in the form of a shell script. So let us consider the case where we put these two programs in the form of a shell script.

(Refer Slide Time: 07:42)

```

Terminal: Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms

awner :1.5 /org/gtk/gvfs/exec_spaw/3
mjsraman 4822 1929 0 Dec26 ? 00:00:00 /usr/lib/gvfs/gvfsd-dnssd --spaw
ner :1.5 /org/gtk/gvfs/exec_spaw/7
mjsraman 5954 1929 0 Dec26 ? 00:00:01 /usr/lib/x86_64-linux-gnu/notify
-osd
mjsraman 24855 1929 0 16:59 ? 00:00:02 /usr/bin/nautilus --gapplication
-service
mjsraman 24881 1929 0 16:59 ? 00:00:00 /usr/lib/libreoffice/program/oos
plash --impress file:///home/mjsraman/IITMadras/Shell%20Programming%201.ppt
mjsraman 24901 24881 3 16:59 ? 00:01:53 /usr/lib/libreoffice/program/sof
fice.bin --impress file:///home/mjsraman/IITMadras/Shell%20Programming%201.ppt -
-splash-pipe=5
mjsraman 25040 1929 0 17:02 ? 00:00:13 /usr/lib/gnome-terminal/gnome-te
rminal-server
mjsraman 25047 25040 0 17:02 pts/1 00:00:00 bash
mjsraman 27258 1929 0 17:52 pts/1 00:00:00 bash
mjsraman 27309 27258 0 17:52 pts/1 00:00:00 sleep 5
mjsraman 27311 25047 0 17:52 pts/1 00:00:00 ps -eaf
mjsraman 27312 25047 0 17:52 pts/1 00:00:00 grep --color=auto mjsraman
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 4
job3
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 2

```

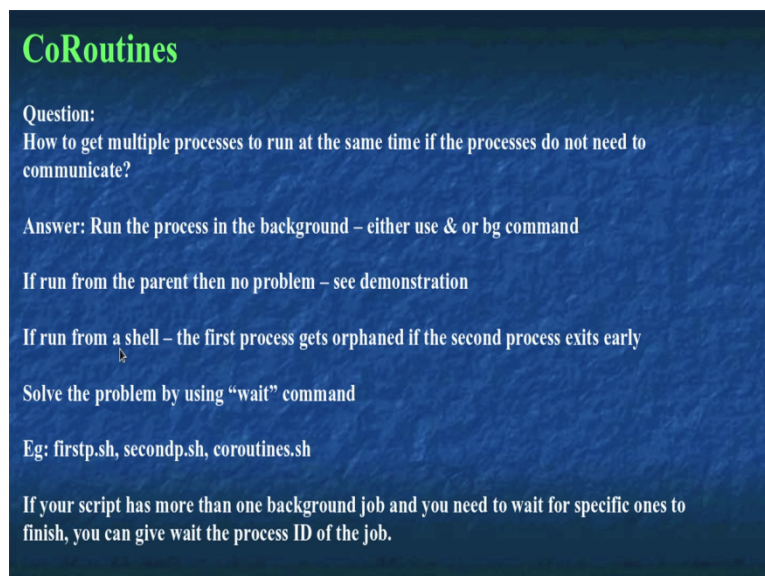
So something like this ok. So in this case what happens is when I run this program this coroutine dot sh program what will happen is that I am trying to put the first program in the background ok and then I am trying to put the second program in the foreground so everything looks fine here so what is happening is that so when I run this program ok so the

coroutine dot sh is working. So if you look at this both the programs have started working and it has come out ok so since the second program has exited it has come out ok and now let us try to see jobs ok so there are no background jobs, so look at this there are no background jobs ok but still your program is running. So what is happening is that it says there are no background jobs but still there program is running so why does this happen?

It is because when I executed this Coroutines dot sh ok this Coroutines dot sh, the Coroutines dot sh had a process Id and that spanned two processes ok the first process was first p dot sh and the second process was second p dot sh ok. So because of which so let me run this program again this Coroutines dot sh so it created two processes ok so if you look at this so it has created two process so you can see two bash shell ok. so it has created one as sleeping for 5 so it has created two processes.

Now what is happening is that so if you look at this now the first process has died then there is the second process that is still going on and on and on ok, and what is happening is that the second process you are not able to see using the job command because you did not execute this from the current shell so the current shell is already died because the current program which invoke this it already died. Because of which you are not able to see using a job, ok so essentially then if the current shell has died that means the Coroutines dot sh has died then how it that your program is still working is.

(Refer Slide Time: 10:19)



CoRoutines

Question:
How to get multiple processes to run at the same time if the processes do not need to communicate?

Answer: Run the process in the background – either use & or bg command

If run from the parent then no problem – see demonstration

If run from a shell – the first process gets orphaned if the second process exits early

Solve the problem by using “wait” command

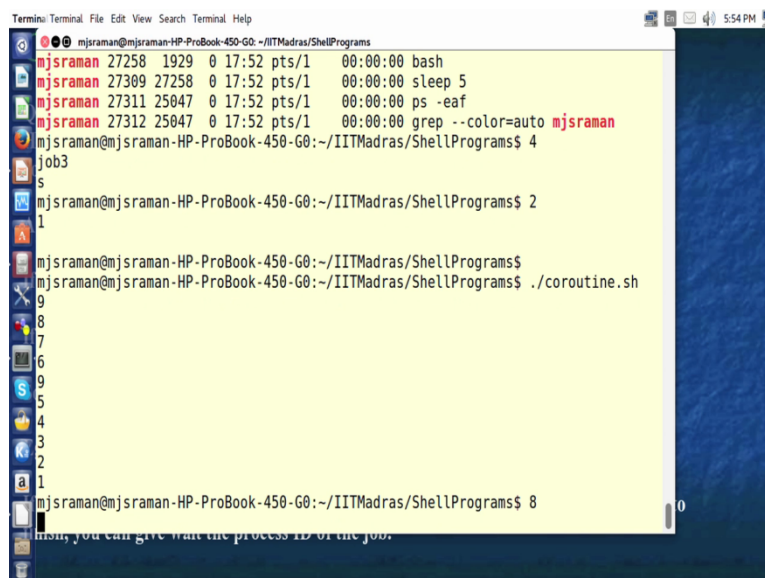
Eg: firstp.sh, secondp.sh, coroutines.sh

If your script has more than one background job and you need to wait for specific ones to finish, you can give wait the process ID of the job.

So here is one of the things that we know ok. So if you run the program from the shell ok for example if you run the example from the shell then as soon as the second program gets over that is second p dot sh gets over what is happening is that the first program still runs in the background. Now how is it possible that your Coroutines program has already existed but the program that created the first p dot sh has exited but then the first p dot sh is still running so what happens is that the first p dot sh will actually get orphaned then the second process exits early ok. That is why when you type jobs command you are not able to see that process.

Now the question is how do I know or what do I do to ensure that I come back to my shell only after the background process as well as the second process that I start ends?

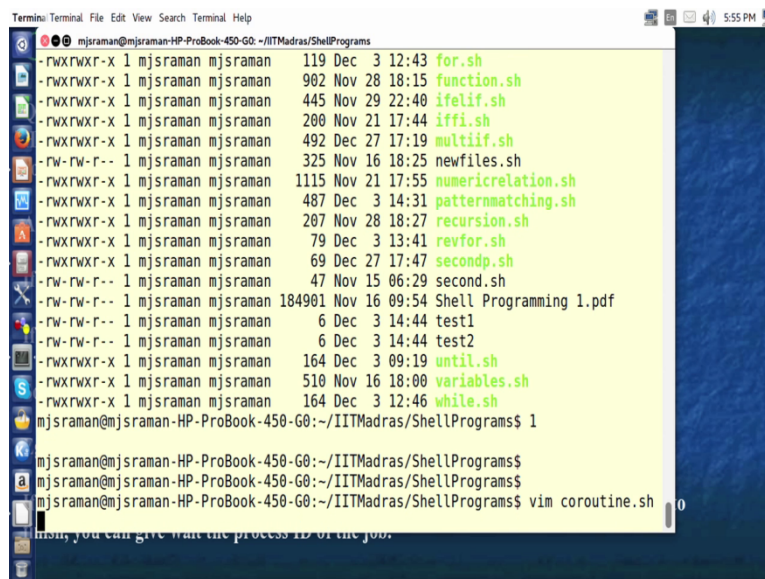
(Refer Slide Time: 11:27)



```
Termin@ Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
mjsraman 27258 1929 0 17:52 pts/1 00:00:00 bash
mjsraman 27309 27258 0 17:52 pts/1 00:00:00 sleep 5
mjsraman 27311 25047 0 17:52 pts/1 00:00:00 ps -eaf
mjsraman 27312 25047 0 17:52 pts/1 00:00:00 grep --color=auto mjsraman
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 4
job3
5
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 2
1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./coroutine.sh
9
8
7
6
5
4
3
2
1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 8
0
```

So what I essentially say is that when I give this command Coroutines dot sh it should not exit immediately after the second program inside Coroutines dot sh exits. So in this case for example the first and the second programs are running and look at this at this point the second program has exited therefore I have come back to the shell you know I am able to type any command that I want and I get the result at the same time there is something that is running at the background.

(Refer Slide Time: 11:58)



```
Termin@ Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
-rwxrwxr-x 1 mjsraman mjsraman 119 Dec 3 12:43 for.sh
-rwxrwxr-x 1 mjsraman mjsraman 902 Nov 28 18:15 function.sh
-rwxrwxr-x 1 mjsraman mjsraman 445 Nov 29 22:40 ifelif.sh
-rwxrwxr-x 1 mjsraman mjsraman 200 Nov 21 17:44 iffi.sh
-rwxrwxr-x 1 mjsraman mjsraman 492 Dec 27 17:19 multiif.sh
-rw-rw-r-- 1 mjsraman mjsraman 325 Nov 16 18:25 newfiles.sh
-rwxrwxr-x 1 mjsraman mjsraman 1115 Nov 21 17:55 numericrelation.sh
-rwxrwxr-x 1 mjsraman mjsraman 487 Dec 3 14:31 patternmatching.sh
-rwxrwxr-x 1 mjsraman mjsraman 207 Nov 28 18:27 recursion.sh
-rwxrwxr-x 1 mjsraman mjsraman 79 Dec 3 13:41 revfor.sh
-rwxrwxr-x 1 mjsraman mjsraman 69 Dec 27 17:47 secondp.sh
-rw-rw-r-- 1 mjsraman mjsraman 47 Nov 15 06:29 second.sh
-rw-rw-r-- 1 mjsraman mjsraman 184901 Nov 16 09:54 Shell Programming 1.pdf
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test1
-rw-rw-r-- 1 mjsraman mjsraman 6 Dec 3 14:44 test2
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 09:19 until.sh
-rwxrwxr-x 1 mjsraman mjsraman 510 Nov 16 18:00 variables.sh
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 12:46 while.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim coroutine.sh
```

Now once the Coroutines dot sh exits the first p which is running now gets orphaned and it is running with its parent shell the Coroutines parent shell so it is working at the background but then I do not want this to happen I want to come to the shell and I should be able to type any command only after firstp and secondp gets over. So what I do is so in this case we have to use command called wait and what it does is so if you I will edit this file called Coroutines dot sh.

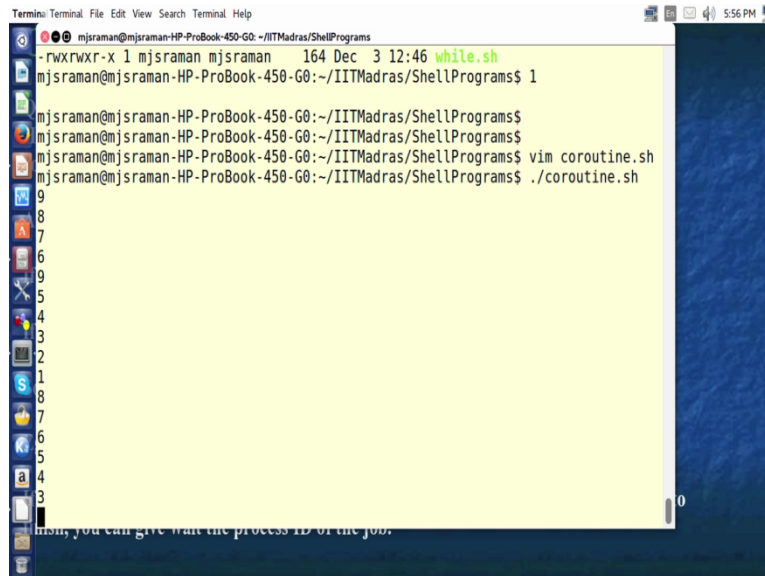
(Refer Slide Time: 12:30)



```
Termin@ Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
1 ./firstp.sh &
2 ./secondp.sh
3 wait
-- INSERT --
3,5 All
```

So in this Coroutines dot sh you see I am doing the same thing but then what I do is I type the command called wait.

(Refer Slide Time: 12:40)

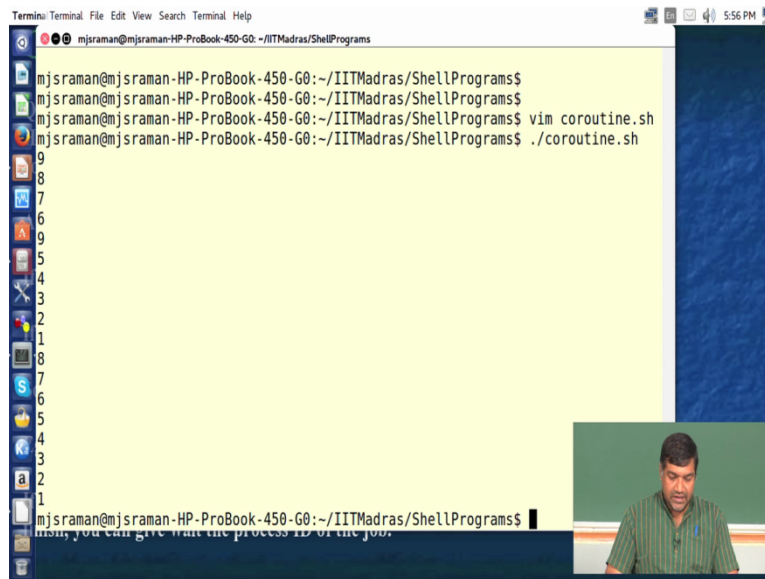


```
Termin@ Terminal File Edit View Search Terminal Help
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
-rwxrwxr-x 1 mjsraman mjsraman 164 Dec 3 12:46 while.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ 1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim coroutine.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./coroutine.sh
9
8
7
6
5
4
3
2
1
0
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
```

Now if you type the command called wait so let us see what happens So I run this coroutine dot sh now you see the first program has started the second program has also started and let us see what happens, so now the second program is going to end and still you have not got the prompt.

So look at this you have not got the prompt, so what is happening now is your shell is actually waiting for the background job to complete so actually if you look at this the first program is still running we have 8,7,6,5,4,3,2 and 1 and only after that you will get back the shell prompt. So in this case now can you get this effect by normal shell programming I will try I leave it as an exercise try to find out whether you are able to do it or not.

(Refer Slide Time: 13:35)



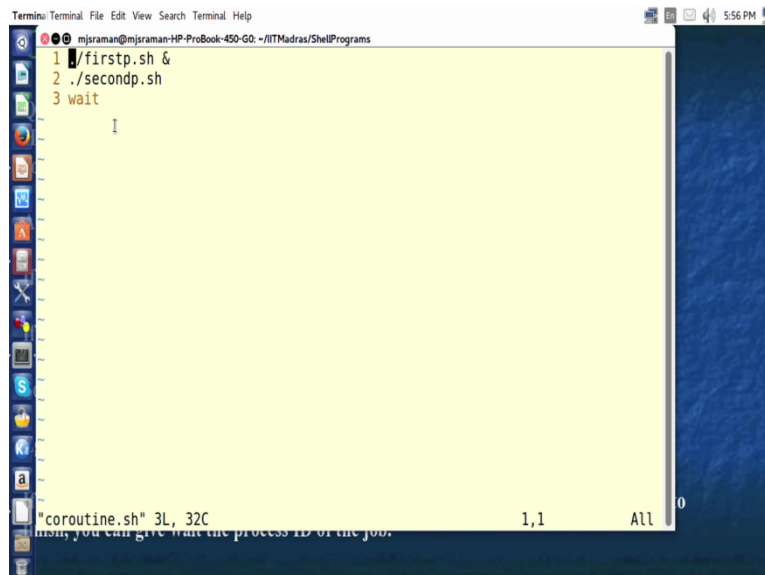
A terminal window titled 'Termino Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help). The window shows the following commands and output:

```
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ vim coroutine.sh
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$ ./coroutine.sh
9
8
7
6
5
4
3
2
1
8
7
6
5
4
3
2
1
mjsraman@mjsraman-HP-ProBook-450-G0:~/IITMadras/ShellPrograms$
```

The terminal output shows a list of numbers from 9 down to 1, followed by another list from 8 down to 1. The prompt returns to the shell. A small video inset in the bottom right corner shows a man speaking.

See now we have got back the shell, so what this wait does is that so if you let us go back to the code.

(Refer Slide Time: 13:42)



A terminal window titled 'Termino Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help). The window shows the following commands and output:

```
mjsraman@mjsraman-HP-ProBook-450-G0: ~/IITMadras/ShellPrograms
1 ./firstp.sh &
2 ./secondp.sh
3 wait

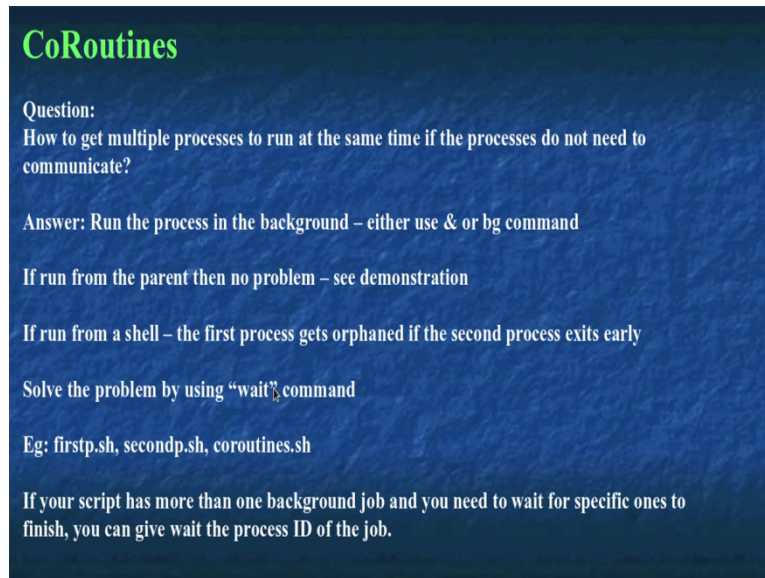
```

The terminal output shows the execution of two scripts in parallel, followed by a wait command. The prompt returns to the shell. A status bar at the bottom shows the file 'coroutine.sh' with line 31, column 32C, and a cursor position of 1,1. The status bar also shows 'All' and '0'.

So what this wait does it that it executes this first program and second program as Coroutines but then waits until all these programs are completed and then it exits. So if you remember in this case we prevent the first guy or the first program from becoming an orphan. So you would have heard about the word orphan that is when the parent process gets killed or exits

then the child process becomes part of the system. So in this case we prevent the first process from becoming an orphan by putting the wait command.

(Refer Slide Time: 14:32)



CoRoutines

Question:
How to get multiple processes to run at the same time if the processes do not need to communicate?

Answer: Run the process in the background – either use & or bg command

If run from the parent then no problem – see demonstration

If run from a shell – the first process gets orphaned if the second process exits early

Solve the problem by using “wait” command

Eg: firstp.sh, secondp.sh, coroutines.sh

If your script has more than one background job and you need to wait for specific ones to finish, you can give wait the process ID of the job.

Now if you look at this so if you you can solve the problem of coming back to you at shell ok before all the programs are completed by putting the wait command. So the other thing that you can do is that in this wait if you can give the process ID then you can say if you have skipped as more than one background say for example I have firstp secondp is also background third p is also background and then fourth p is running in the foreground.

Now suppose I want all of them to complete and then you want to come back to the shell then I can just type wait at the end of the shell script. Now if I want to only first p to be completed then I can give the wait for the specific process ID and then find out how to exit. Now in this way we will ensure that the all the processes are completed before I go to the next execution of the shell script see this is the idea ok. So sometimes if you use this if you do not use wait what happens is that the current shell goes off one process becomes orphan and capturing output of the process and processing it you could get some errors. So in order to avoid it what you do is when I say wait then it waits for all the process to finish upto that point and then it continues execution after this point.

(Refer Slide Time: 15:55)

What are the advantages and disadvantages of CoRoutines ?

A computer with only one processor (CPU), you might have performance advantage.

We characterize a process in terms of how it uses system resources in three ways: CPU, I/O, Interactive

Processes differ with respect to these three criteria, the more advantage there is in running them simultaneously.

However, if two processes use resources in similar ways, sometimes it may even be less efficient to run them at the same time

(A small video inset in the bottom right corner shows a man in a green shirt speaking with his hands.)

So now what are the advantages and disadvantages so why should I talk about this or why should I know about this, ok? So if you use Coroutines ok the advantage is if your computer has only one processor let us say you might actually have a slight improvement in performance whereas in executing them serially, ok? So this is sort of a parallel processing that we are looking at with shell itself. Now if you look at this and will this be really useful? yes, because not all the shell commands or commands that you run on UNIX take the same amount of CPU or input output or interactive so your whole performance of any program depends on what is its usage of CPU?

What is its usage of input output? what is its usage of instructiveness? Now any process can differ with respect to these three criteria and therefore you can running them simultaneously will you be able to really able to time share between two processes and actually sometimes it may so happen that if you want to run two independent process you can run them together by using these Coroutines. But ensure that if you want all the processes to complete and then proceed you use something like wait ok?

Can these Coroutines have disadvantages? yes probably, what could happen is that if there are two process that use resources in the similar way ok? Remember these two process should be independent of each other because if they are dependent then one output should be given to the others input so that is by typing. If they are independent of each other then you can definitely get some kind of performance improvement of course it depends on can you really

guarantee performance improvement in most of the cases yes in some cases probably no, then they use the same resources or same CPU at the same time etc and then it could slow down but this definitely can be used to speed up your shell scripts.