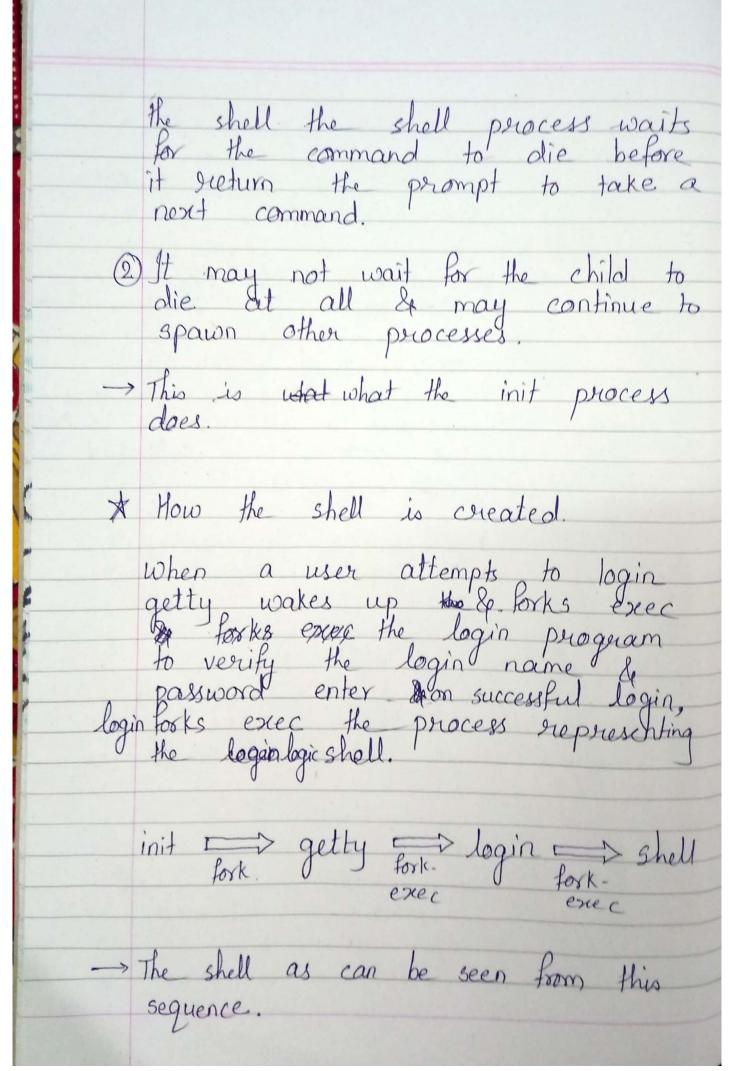
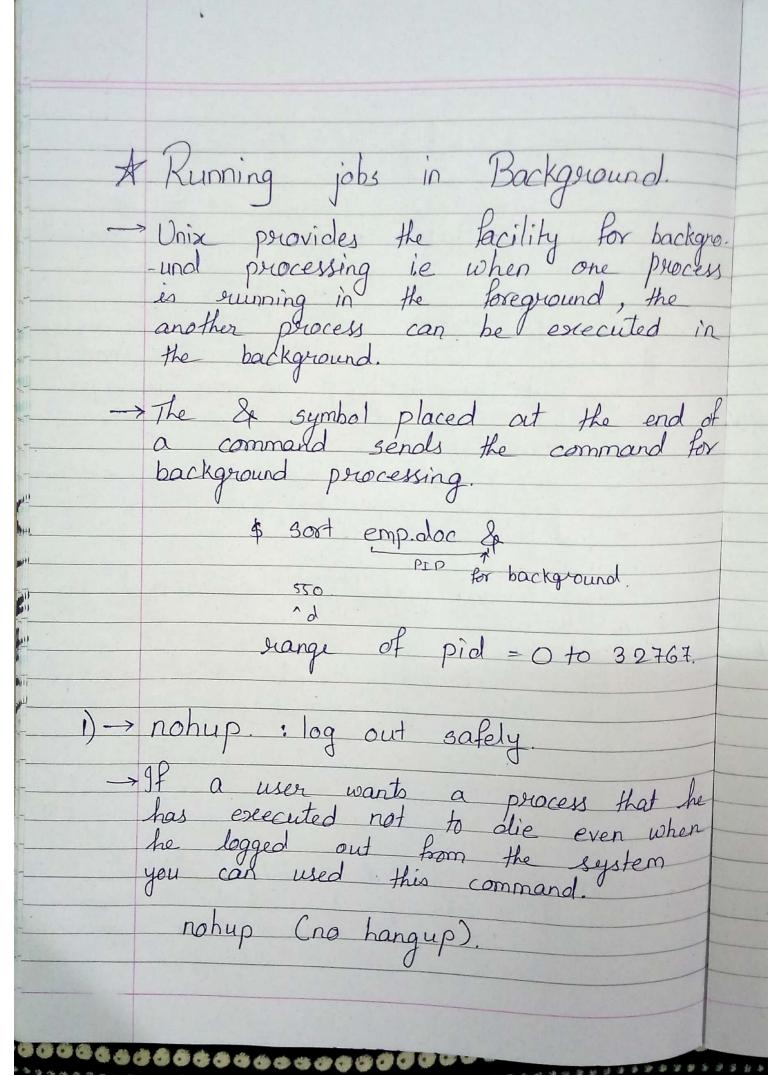
\* Parents & Children. -> Just as a file has a parent every process also has one, This parents into itself is another process & a process porn from it is said to be its child. -> When you run the command cat emp. 1st from the keyboard a process represent -ing the cat command is started by the shell process. -> Since every process how a parent you cann't have orphaned process, \* Would or not would Two different attitudes that can be taken by the parent towards its 1) It may wait for the child to clie so that it can spanwon the next process. The death is informed by the Kernel to the parent. When you execute a command from



-> inits goes off to sleep, waiting for the death of its children. The other processes getty be login had extinguish them self by overlap when the user logs out the shell is killed of the death is intimated to init. - Inits then wakes up & spawns another getty for that line to monitor the next login. > Getty is unix program lunning on a host computer that manages physical or virtual terminals. Then heit detects a connection it prompts for a username & runs the login program to authenticate the user. \* Internal & External commands. -> From the process point of view shell recognizes three types of commands 1) External commands. The shell creates a process for each of this commands that it executes while

remaining their parent e.g cat comm- and.
Dell scripts.
The shell executes this scripts by spouring another shell which then executes the commands listed in the script.
The child shell becomes the parent of the commands that featured in the script.
3 Internal commands,
The shell has a number of built in commands like cd & echo both don't generate a process and our executed directory by the shell.
Similarly valriable assignment of the Statement x=5 for instance doesn't generate a process either.

* '	Brocess states & Zombies.
	A process after creation is in the runnable state before it acc actually
Li k	Thile the process is sunning it may be invoke a disk i/o operation when it has nothing to do except wait for the i/o to complete.
	the process then moves to the sleeping state to be woken up when the ilo operation is over.
2	Process can also be: suspended by a key.
→ °	Processes whose parents don't wait for that their death moves to be zombies state.
	When the process dies it eximmediately moves to the zombie state.
→ → →	It remains in the state until the parent picks up the child exceed exit. I status from the process table and entry.
600	



syntax
\$ nohup 2000 mand > 8p
eg \$nohup sort emp.doc&
55
L'output on [nohup.out]
2) \$at.
This command is used to execute the specified unix command at future time
syntax:
at <fime> <commands></commands></fime>
And od. (control d)
e.g \$ at 12:00 echo "LUNCH BREAK"
* Keywords.
1) now 6) hours
2) noon 7) days
3) midright 8) weeks 4) today 9) months
4) today 9) months 5) tomorrow 10), years

e.g \$at 12:00 + 1 day } Gives echo "LUNCH BREAK" Jremainder tomorrow on 12:00
3) atq
→ atg command is used to list the jobs submitted by you atg gueue.  → This command list the job no., schedu - led date of execution.
e.g \$ cit q hows queue worrante  7 2002-12-16 12:00 a bmi 8 2002-12-17 12:00 a bmi 9 2003-01-22 13:00 a bmi 4) \$at year.  4) \$at queue.
e.g. darm 8  O/P  7 & & 9 / of  Job no. 7 & & 9 is display -ed.

5)\* shortch This command is used to execute the specified command when the system load permits Crohen CPU becomes nearly force). Syntax & batch <commands> e.g & botch a.sort to. sort 6) \$Kill. -> If you want a command to terminate pre-maturely press ctol d. This type of interrupt characters does not affecting background processes because the background processes are protected by the shell from these interrupt signals. - This kill command is used to terminate the background process.

$\rightarrow$	Guadan duil Cair land 7 1900
	Syntax \$ kill [signal Number] < P90> e.9 \$ kill 551
$\rightarrow$	Bu dolla He will accome and will the
	signal number 15 to terminate a process but some programs like
	signal number 15 to terminate a process but some programs like login shell simply ignore, this signal of interruption & continue execution normally.
	In this case you can use the signal number 9. Often referred as
	signal number 9. Often referred as
	(-9 for login process kill).
	Syllabus
	9.1 Process basics
	9.1.1 The shell process
	9.1.2 Parents & children
	9.1.3 Wait or not wait
	9.2 Ps (process status) 9.2.1 Ps options.
	1.2.1
	9.3 System processes.

0	9.4.1 How the shell is created.
9	9.4.1 How the shell is created.  9.4.1 Internal & external command
9	1.6 Process states & zombies
9	9.7.1 &: No loggin out 9.7.2 nohup: log. out safely
• 9	1.8 Nice: job execution with low priority
9	1.9 Killing processes with signals 9.9.1 Kill: Premature termination of process.
9	1.11 at & batch: execute later.  9.11.1 at: one fine execution  9.11.2 batch: execute in batch queue.
7)	Hire frice Creduce the priority).
-> 1	Inix offers this command which is used with the (84) operator to reduce the priority of jobs.

	A higher nice value implies a lower priority.
	Nice reduces the priority of any process thereby rising its Nice value.
	We can also specified the a Nice value explicitly with -n option.
No. of the case of	e.g \$ nice we -1 uxmanual &
N PA	eg \$ nice -n 5, wc -l uxmanual & 1-19 range.
	(Mosie priority more delay)
pl!	
-57	