

Text processing using awk command in linux   awk patterns & actions   BEGIN,END,NR, FS,OFS,RS,ORS	<a href="https://www.youtube.com/watch?v=X-3AXAsU3Wo">https://www.youtube.com/watch?v=X-3AXAsU3Wo</a> YT: Pedagogy	
Awk generally has 2 parts, the pattern part and the action part and these are written inside single quote		
Action part is written inside curly braces		
awk ' {print } ' inputFile	No pattern just action to print the each entire line(record) in the file	
awk '/manager/ {print \$0}' inputFile	In this case, the pattern is manager. Print each entire line that has the string "manager" anywhere in the line.  awk '/manager/ {print \$0}' inputFile awk '/manager/ {print}' inputFile  Both are same	
RS	Record separator (line separator)	
FS	Field separator (First word is the first field and so on)	
awk '/manager/ {print \$1}' inputFile	Print the 1st field of each line if the line has the string manager somewhere in it.	
awk '/manager/ {print \$1, \$3}' inputFile	Print the 1st and the 3rd field of each line containing the string manager.	
awk '/^p/ {print \$0}' inputFile	Print all lines that start with character p (xor operator) <b>(Regex)</b>	
awk '/^[pvg]/ {print \$0}' inputFile	Print all lines that start with either p or v or g	
<b>Giving conditions based on some field</b>	From timestamp 4.09	
awk '\$2 ~ /clerk/ {print \$0}' inputFile	Print all those lines which has the 2nd field containing the characters clerk (Note: the 2nd field should contain the string clerk. So, even if the 2nd field value is clerks with an extra s then also this line will qualify to be printed)	How to print only those records where 2nd field value is exactly clerk? Ans: awk '\$2 == "clerk" {print \$0}' inputFile This will look for exact match in the 2nd field.
awk '\$2 !~ /clerk/ {print \$0}' inputFile	Print all those lines whose 2nd field does not contain the string clerk (Opposite of the above result)	
<b>Using comparison operator in the awk pattern</b>	From timestamp 5.45	
awk '\$5 > 1500 {print \$0}' inputFile	Print those lines where the value of 5th field is greater than 1500	
<b>Combining multiple patterns</b>		
awk '\$5 > 1500 && \$5 !~ /sal/' inputFile	print all those lines where the 5th field value is > 1500 AND the 5th field doesn't contain the string sal	
<b>Specifying the range in the pattern</b>		
awk '\$4==50000,\$5==500 {print \$0}' inputFile	Perform the action (in this case print) from the line where 4th field value=50000 and stop printing at the line where 5th field value is 500	
<b>Specifying BEGIN and END</b>	BEGIN and END action are processed just once.	
awk 'BEGIN {print "awk started"} {print \$0} END {print "awk ended"}' inputFile		
awk 'BEGIN {print "awk started"} /manager/ {print \$0} END {print "awk ended"} ' inputFile		
<b>NR variable</b>	Number of records processed	
awk ' {print NR, \$0}' inputFile		
awk '/manager/ {print NR, \$0}' inputFile		
<b>NF variable</b>	Number of fields in the record	
awk '/manager/ {print NR, \$0, NF} ' inputFile		
awk '{print \$NF}' inputFile	Print the last field of each line (value of last field)	
<b>OFS, ORS, FS</b>	Output Field Separator, Output Record Separator, Field Separator(Input Field Separator)	
awk 'BEGIN{OFS="@"} {print \$1, \$2} ' inputFile	Note: You must have comma inside the print action for OFS to kick in.	
<b>Awk can also process the output coming from another command</b>	From timestamp 17.41	
awk 'NR==4,NR==8 {print \$0} ' inputFile   awk '\$3 ~ /sales/ {print \$0}'	For example, I want to process only line numbers 4 till 8 and print those lines which have the value sales in field number 3 (piping)	
awk 'NR==4,NR==8 {print \$0} ' inputFile   awk '\$2 == "clerk" {print \$0}'		
<b>Using awk with some script file using -f option</b>		

<pre>{ if (\$NF &gt; 500) print \$NF else print "hi" }  BEGIN{ print "Awk is awesome" } { myVar=99 if(\$NF &gt; 500 &amp;&amp; \$5 != "salary") print \$NF+myVar else print "line: ", NR, " did not meet the condition" } END{ print "peace to all" } </pre>		
<b>Using inbuilt function with awk</b>		
awk '{print toupper(\$2)}' inputFile	The value in field 2 in every line will be printed in uppercase.	
awk '{print toupper(\$2),length(\$3)}' inputFile	Get the length of the string in field 3 of every line	