

## Re: AWK problem need help

---

*Source:* <http://unix.derkeiler.com/Newsgroups/comp.unix.shell/2007-06/msg00463.html>

---

- *From:* Ed Morton <[morton@xxxxxxxxxxxxxxxx](mailto:morton@xxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 19 Jun 2007 06:49:32 -0500
- 

Akhil wrote:

Hi,  
I am trying out an "AWK" script, to search and modify some pattern in text file.  
The script is as follows:

```
NR<100 {
if ( ( $6 ~ /^(x|X)/ ) || ( $4 ~ /^(x|X)/ ) ) {
printf ("PAT:%s\n", $prev)
print NR, $0
z = split($prev, lastline, " ")
if ( lastline[0] ~ /^\\// ) {
print NR >> "bin/output.atp"
}
if ( lastline[0] ~ /repeat/ ) {
print NR, NR >> "bin/output.atp"
} else {
print NR, NR, NR >> "bin/output.atp"
}
} else {
print $0 >> "bin/output.atp"
prev = $0
}
}
```

My observation here is:

In else part (i.e. when I dont get the regx match, I am trying to store the line read) in a variable, and I wned to use it later. But when I try printing the variable "prev" next time when I have "regx" match, it displays the current line i.e. \$0 itself

Please suggest some alternative to store the previously read line.

Akhil

## Re: AWK problem need help

awk is not shell. To print the value of an awk variable is "print var", not "print \$var". The latter would print the field at whatever position "var" contains, and the whole record if var is zero or unset.

Also, you don't need the " " as the final argument to split() as that's the default, and awk arrays start at 1, not 0 so "lastline[0]" will always be empty.

Finally – are you sure you know what "print >> " does in awk? Again, awk is not shell...

Ed.

.