

grep, find, cut, sort, uniq, wc, split, curl, awk, xargs, sed, tr

1. grep/egrep/fgrep

-i (ignore_case); -e (regexp); -v (invert); -o (only_match); -n (line_number); --color;

```
cat G2JSON.log | grep -o "CXP9024418\_89\R.."; grep -n --color "rerun" *
```

2. find

-maxdepth n; -mindepth n; -name "x"; -regex "x"; -iname "x" (case insensitive); -iregex "x" (case insensitive);

```
find /local_stg/lterbsFtp_up/up -noleaf -maxdepth 1 -mtime +90 -name "CXP102051*" | xargs  
rm -rf; find . \( -name "plugins" -prune \) -o \( -regex "\(.*\)\.html" \)
```

3. cut

-b n (bytes); -c n (characters); -d x (delimiter) -f n (fields);

```
who | cut -b -3,3- ; who | cut -b 1-2,4 ; who | cut -d "\" -f 1
```

4. sort

-u (unique); -r (reverse);

```
find /local_stg/lterbsFtp_up/up -maxdepth 1 -type d -name "CXP9024418_8*" | sort -r | grep -o  
"CXP9024418\_89\R[0-9][A-Z]*"; w | sort -u
```

5. uniq

-d (repeated); -u (unique); -c (count);

```
w | uniq -d; who | uniq -u; who | uniq -c
```

6. wc

-l (lines); -w (word); -c (bytes) -m (chars);

```
who | wc == who | wc -l,-w,-c/-m
```

7. split

-b (bytes); -l (lines); -n (file_number);

```
split -n 3 test.txt prefix; cat prefix* > file
```

8. curl

--date "x"; -o "file"; -v;

curl -v -o /G2JSON.log

http://xx?product_number=CXP9024418_89&increment_number=FT_89.9&confidence_level=2&verdict=SUCCESS&latest=1; curl -v -k --noproxy 20.1.40.23 --data "DoBoardRestore=BoardRestore" https://20.1.40.23/cgi-bin/aicGui:post

9. awk

awk -F 'x' 'BEGIN {action} /pattern/ {action} END {action}';

ps -A -o stat,ppid,pid,cmd | grep -e 'bjenb03' | awk '{print \$3}' | xargs kill -9; last -n 5 | awk -F ' ' 'BEGIN {print "AWK started"} /pts\2/ {print NR,"NF","\$1","\$2} END {print "AWS ended"}'

10. xargs

-l 'x' (replace-str); -i (default {})

ls \$testupname | ls CXP9024418_89-R2E | grep -o 'CXP9024418_89\R.*zip\..*' | xargs -i mv \$testupname/{}/home/lterbsbj/ejqizng/G2_UP_temp/; ls \$testupname | ls CXP9024418_89-R2E | grep -o 'CXP9024418_89\R.*zip\..*' | xargs -l [] mv \$testupname/{}/home/lterbsbj/ejqizng/G2_UP_temp/; ls | xargs -i mv {} {}.bak

11. sed

sed 's/pattern/replacement/'; sed 's/pattern/replacement/g'; sed '/pattern/d'; sed 'i\str';

#replace string

cat sed.txt | sed "s/\./\n/g"

#extract replace string

cat sed.text | sed "s/^(.\\.)/\1/g"; cat sed.text | sed "s/.(\\.)\$/\1/g"; cat sed.text | sed "s/([A-Z])\1/g"; cat G2JSON.log | grep -o "CXP9024418_89\R[0-9][A-Z]*\.zip" | sed "s/(CXP9024418_89\R[0-9][A-Z]*)\.zip/\1/g"

#add string

cat sed.text | sed "s/xvcxv/&haha&/g"; cat sed.text | sed 's/\-.*\-/&/'

#delete string

```
cat sed.txt | s/[\r\n]//g; cat sed.txt | s/s+$//g;
```

#add line

```
cat ls.txt | sed "2i\hello"; cat ls.txt | sed "1i\hello"; cat ls.txt | sed '$a\hello'
```

#delete line

```
cat sed.txt | sed "s/^./g"; cat sed.txt | sed "s/.$//g"
```

```
cat sed.txt | sed 2,"$d"; cat sed.txt | sed '2,$d'; cat sed.txt | sed "/^$/d"; cat sed.txt | sed  
"/^\.$/d"; cat G2JSON.log | grep -o "CXP9024418\_89\R.." | sed 2,'$d'
```

##file edit

```
sed -i '$a\hello' ls.txt; sed -i "1i\hello" ls.txt; sed -i '1i\<p>G2 89.9 FT Track UP List</p>' /local_stg/jenkins/build_conf/up_list_g2.html
```

12. regular expression - metacharacter

- ① \ (escape characters) -> \{\}; \(\)
- ② . (Matches any single character); [] (Matches a single character that is contained within the brackets); [^] (Matches a single character that is not contained within the brackets)
- ③ * (Matches the preceding element zero or more times); ? (Matches the preceding element zero or one time); + (Matches the preceding element one or more times); \{m\} (Matches the preceding element m times); \{m,\} (Matches the preceding element at least m times); \{m,n\} (Matches the preceding element at least m and not more than n times); | (Matches either the expression before or the expression after the operator)
- ④ ^ (Matches the starting position within the string); \$ (Matches the ending position of the string or the position just before a string-ending newline) -> ^\$ (blank line)
- ⑤ \(\) (Defines a marked subexpression); \n (Matches what the nth marked subexpression matched, where n is a digit from 1 to 9)

13. tr

-d (delete); -s (squeeze repeats);

```
echo "HELLO WORLD" | tr 'A-Z' 'a-z'; echo "hello 123 world 456" | tr -d '0-9'; echo "thissss is  
a text linnnnnnne." | tr -s ' sn'; echo 1 2 3 4 5 6 7 8 9 | xargs -n1 | echo ${tr '\n' '+'} 0 ]
```