PRONET

VNU-ITP

UNDP

# TEXT PROCESSING

# TEXT PROCESSING

- Search text in files with `grep` family commands: `grep, egrep, fgrep`
- Modify text file with `sed` (Stream Editor)

# `grep` Command

- Use the `grep` command to search a file for a specified text string

- If found, `grep` prints all lines that contain that pattern to the screen. The `grep` command can be used as a filter with other commands.

- The `grep` command is case sensitive unless you use with –i option

# **grep** Command

- Command Format :

```
grep [option] string filename
```

**Options :**

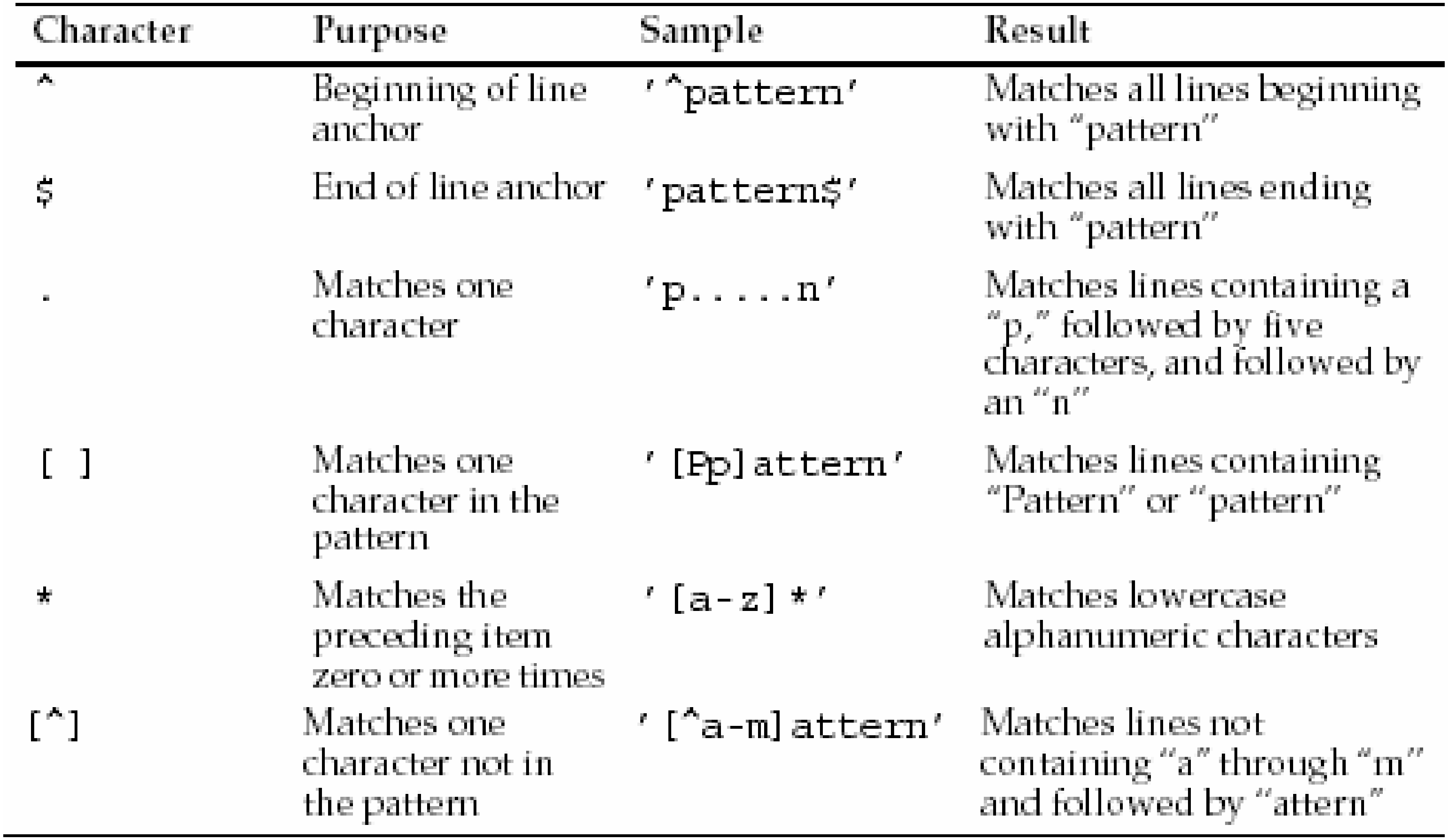
| | |
|---|---|
| **-i** | **Ignore case of string when searching** |
| **-v** | **Search for all lines that do not match string** |

# Example of grep

```
$ grep ‘root’ /etc/passwd
root:x:0:1:Super-User:/:/sbin/sh

$ ls -la | grep -i ’jun 11’
drwxr-xr-x 3 user1 staff 512 Jun 11 13:13 dir4
```

# Regular Expression Metacharacters



| Character | Purpose | Sample | Result |
|---|---|---|---|
| ^ | Beginning of line anchor | '^pattern' | Matches all lines beginning with "pattern" |
| $ | End of line anchor | 'pattern$' | Matches all lines ending with "pattern" |
| . | Matches one character | 'p.....n' | Matches lines containing a "p," followed by five characters, and followed by an "n" |
| [ ] | Matches one character in the pattern | '[Pp]attern' | Matches lines containing "Pattern" or "pattern" |
| * | Matches the preceding item zero or more times | '[a-z]*' | Matches lowercase alphanumeric characters |
| [^] | Matches one character not in the pattern | '[^a-m]attern' | Matches lines not containing "a" through "m" and followed by "attern" |

# egrep Command

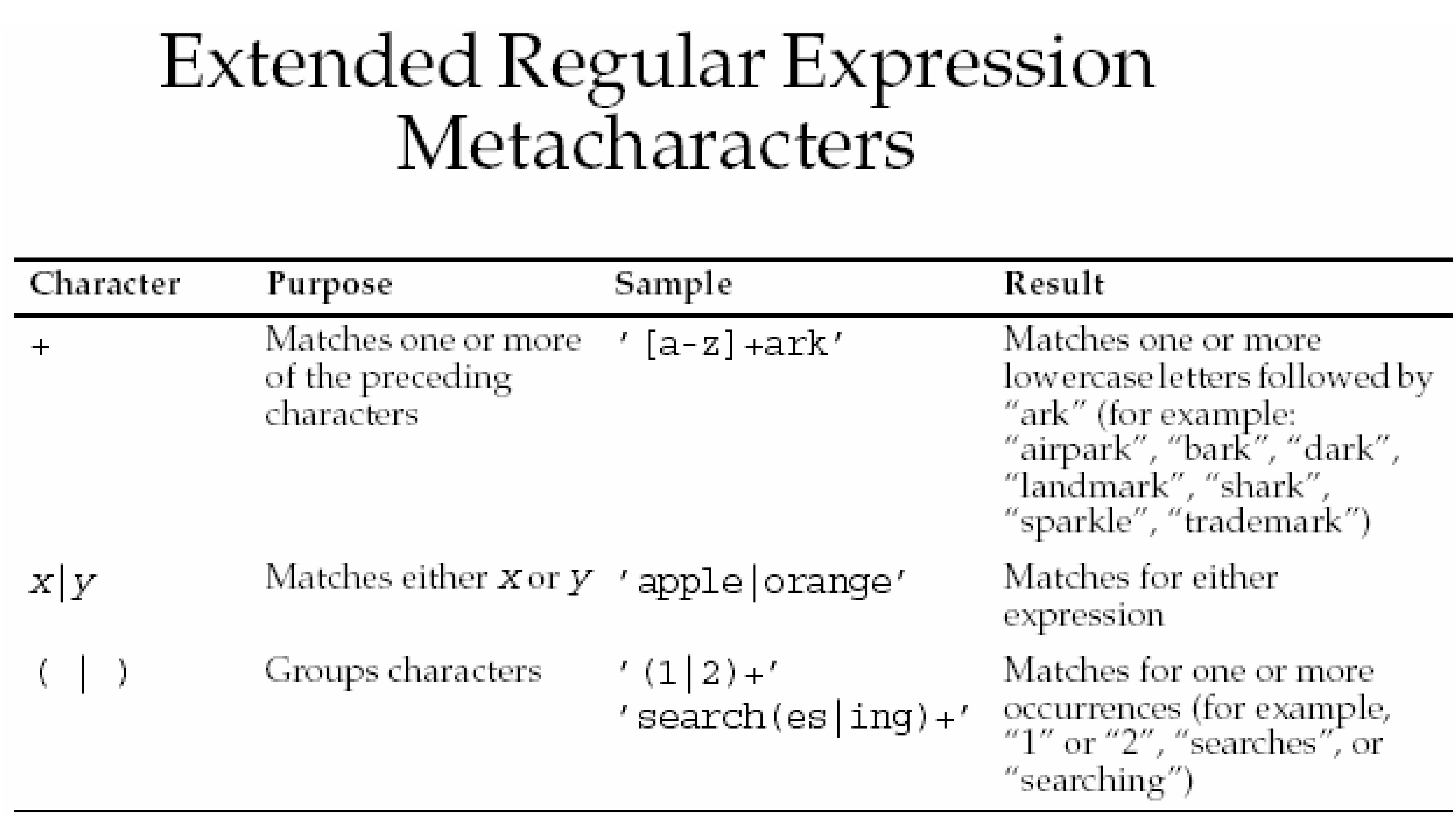
- The `egrep` command searches the contents of one or more files for a pattern using *extended regular expression* metacharacters.
- The `egrep` command uses all the same options as `grep`.
- Command format:

```
egrep [-options] pattern filename(s)
```

PRONET

VNU-ITP

UNDP



# Extended Regular Expression Metacharacters

| Character | Purpose | Sample | Result |
|---|---|---|---|
| + | Matches one or more of the preceding characters | `'[a-z]+ark'` | Matches one or more lowercase letters followed by "ark" (for example: "airpark", "bark", "dark", "landmark", "shark", "sparkle", "trademark") |
| *x*\|*y* | Matches either *x* or *y* | `'apple|orange'` | Matches for either expression |
| ( \| ) | Groups characters | `'(1|2)+'`<br>`'search(es|ing)+'` | Matches for one or more occurrences (for example, "1" or "2", "searches", or "searching") |

# fgrep Command

- The fgrep command searches a file for a pattern expressed as a *fixed* string.
- It differs from `grep` and `egrep` because it regards all characters literally, and it does NOT interpret regular expression metacharacters specified on the command line.
- Use `fgrep` to search for a specific pattern in a file that includes metacharacter symbols.
- Command format:

```
fgrep option(s) pattern filename(s)
```

# Using grep

- List all directories :

```
# ls -l | grep ‘^d’
```

- List all users using `bash` shell when log in :

```
# grep ‘bash$’ /etc/passwd
```

- List files and directories starting by ‘.’ :

```
# ls -l | grep ‘^\.[^.]’
```

# Filters

- Linux traditionally has many utilities that read a stream of characters from stdin, process it in some way.and output their result to stdout. These utilities are called filters

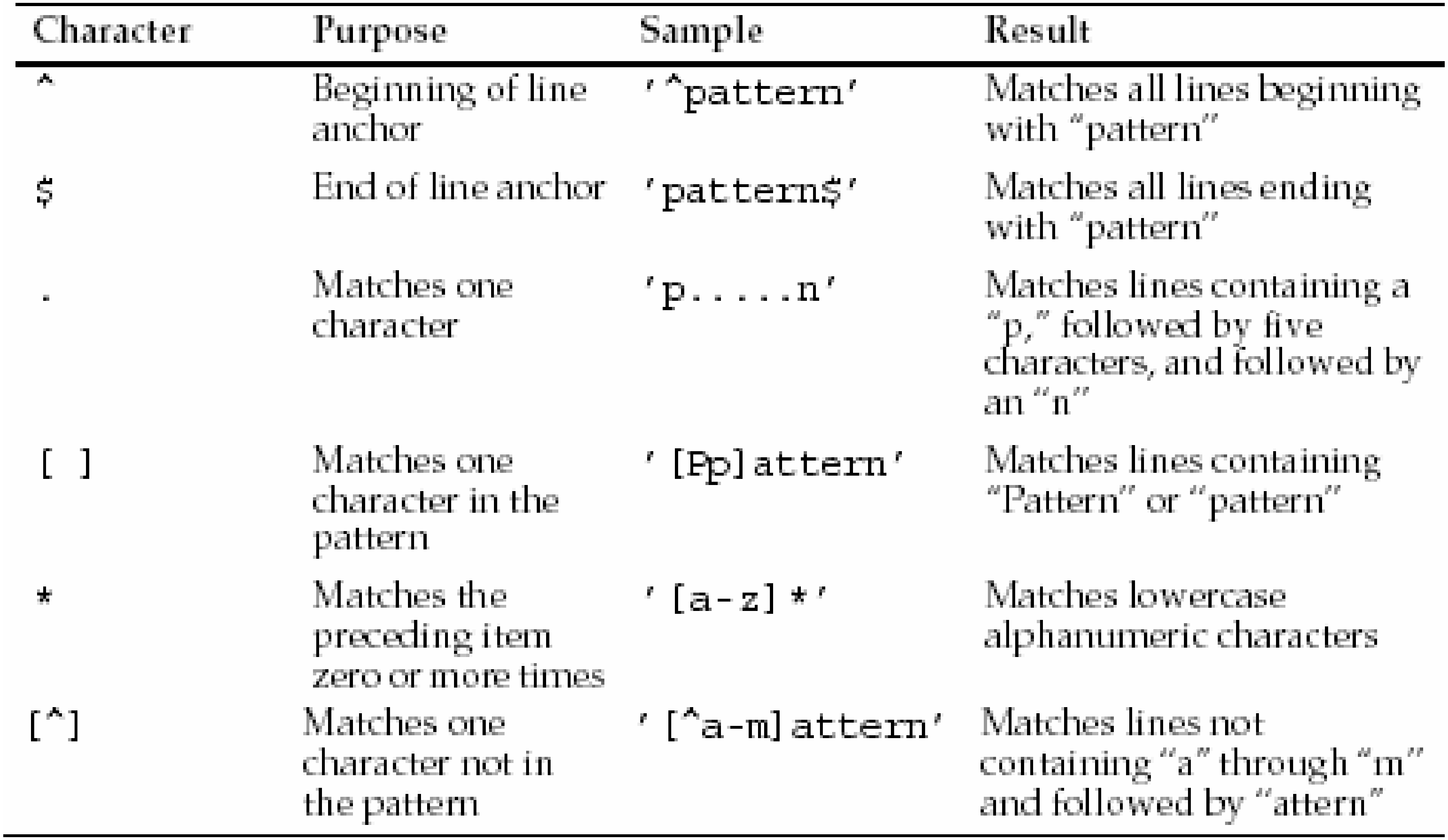
| | |
|---|---|
| cat | concatenate files and print on the standard output |
| cut | remove sections from each line of files |
| sort | sort lines of text files |
| head | output the first part of files |
| tail | output the last part of files |
| wc | print the number of bytes, words, and lines in files |
| nl | number lines of files |

# sed (Stream Editor)

- Use the `sed` program to edit data in files without opening them in an interactive editor, such as vi.
- It allows you to specify edits, or modifications, to a file from the command line and send the output to the screen by default.
- It is best used to make the same changes across multiple files quickly. It is also used to read scripts that administrators need.
- Command format:

```
sed [options] [address] command file...[>newfile]
```

# Regular Expression Metacharacters Used by sed



| Character | Purpose | Sample | Result |
|---|---|---|---|
| ^ | Beginning of line anchor | `'^pattern'` | Matches all lines beginning with "pattern" |
| $ | End of line anchor | `'pattern$'` | Matches all lines ending with "pattern" |
| . | Matches one character | `'p.....n'` | Matches lines containing a "p," followed by five characters, and followed by an "n" |
| [ ] | Matches one character in the pattern | `'[Pp]attern'` | Matches lines containing "Pattern" or "pattern" |
| * | Matches the preceding item zero or more times | `'[a-z]*'` | Matches lowercase alphanumeric characters |
| [^] | Matches one character not in the pattern | `'[^a-m]attern'` | Matches lines not containing "a" through "m" and followed by "attern" |

# sed (Stream Editor)

- Delete lines with the d command:

```
sed ‘/pattern/d’     filename
sed ‘4d’           filename
sed ‘2,5d’         filename
```

- Search and replace :

```
sed ‘s/string1/string2/g’     filename
```

- Multiple edits: use –e option

```
sed –e ‘/linux/d’ –e ‘s/abc/ABC/g’
filename
```

# Exercise

PRONET

VNU-ITP

UNDP