

ASSIGNMENT 3: AWK

CS3423 - Systems Programming

Rocky Slavin & Sam Silvestro

Part A

For this part of the assignment, you will create a **single command** which will take the contents of a `passwd` file (usually found in `/etc/passwd`) and print it in sorted order by the user's last name (that is, their surname, not their username). Normally, you could solve this with the following options on `sort`: `$ sort -t: -k6 /path/to/passwd`

You, however, must solve this problem with the utilities covered in class so far. You may (and should) use `sort`, but you may not use any of its options (e.g., `-k`, `-t`, etc).

Example

Input:

```
1 lkj293:x:1539:1543:Albert Einstein:/home/einstein:/bin/bash
2 kkr590:x:1540:1544:Elvis Presley:/home/presley:/bin/bash
3 nwk409:x:1541:1545:George Washington:/home/washington:/bin/bash
4 yaa265:x:1542:1546:Bruce Banner:/home/banner:/bin/bash
5 yhn211:x:1543:1547:George Harrison:/home/harrison:/bin/bash
6 lfa806:x:1544:1548:Jane Austen:/home/austen:/bin/bash
7 ilo709:x:1545:1549:Walt Disney:/home/disney:/bin/bash
8 rfd576:x:1546:1550:Buzz Aldrin:/home/aldrin:/bin/bash
9 lko889:x:1547:1551:Marie Curie:/home/curie:/bin/bash
10 cfq219:x:1548:1552:J.R.R. Tolkien:/home/tolkien:/bin/bash
11 ncz856:x:1549:1553:Christopher Columbus:/home/columbus:/bin/bash
12 pql747:x:1550:1554:Julius Caesar:/home/caesar:/bin/bash
```

Output:

```
1 rfd576:x:1546:1550:Buzz Aldrin:/home/aldrin:/bin/bash
2 lfa806:x:1544:1548:Jane Austen:/home/austen:/bin/bash
3 yaa265:x:1542:1546:Bruce Banner:/home/banner:/bin/bash
4 pql747:x:1550:1554:Julius Caesar:/home/caesar:/bin/bash
5 ncz856:x:1549:1553:Christopher Columbus:/home/columbus:/bin/bash
6 lko889:x:1547:1551:Marie Curie:/home/curie:/bin/bash
7 ilo709:x:1545:1549:Walt Disney:/home/disney:/bin/bash
8 lkj293:x:1539:1543:Albert Einstein:/home/einstein:/bin/bash
9 yhn211:x:1543:1547:George Harrison:/home/harrison:/bin/bash
10 kkr590:x:1540:1544:Elvis Presley:/home/presley:/bin/bash
11 cfq219:x:1548:1552:J.R.R. Tolkien:/home/tolkien:/bin/bash
12 nwk409:x:1541:1545:George Washington:/home/washington:/bin/bash
```

Script Execution (Part A)

Since the fox machines do not have useful `/etc/passwd` files (no first and last names), you will use the one provided with this assignment. Your submission will include a bash file (`assign3A.sh`) with *exactly one line* in it (you do not need a shebang) and should take the path to the `passwd` file as the first argument. Do not include an `awk` file or any other files besides `assign3A.sh`.

```
$ assign3A.sh /path/to/passwd
```

Part B

For this part of the assignment, you will only use the utilities covered in class so far (primarily `awk`) to create a program for printing user process information. Do not use Python or any programs/utilities not covered in class.

Your program should take the output from `ps -ef` and print the following for each user **having a username matching the abc123 format**:

- Username
- List of commands

After listing statistics for each user, the program should print the following information for all users having a username matching the abc123 format:

- Line with earliest start time
- Line with latest start time

Do not use `sed`, Python, or any other languages/utilities not covered in class.

Example

The example below is an excerpt from the `ps -ef` command which your program should be able to take as input. Note that if a process began execution on a previous calendar day, its `STIME` value will not be in the usual “hours and minutes” format, but rather in “month and day” format. This should be accounted for properly, and thus a simple text/numerical comparison will not suffice.

Input:

	UID	PID	PPID	C	STIME	TTY	TIME	CMD
1	adz110	5344	5334	0	08:47	pts/2	00:00:00	bash
2	dmq292	6908	6854	0	Jun04	pts/1	00:00:00	bash
3	adz110	7227	7150	0	Jul11	pts/9	00:00:00	who
4	erg474	7466	7461	0	08:54	pts/10	00:00:00	ls
5	dmq292	7966	7960	0	Jun04	pts/13	00:00:00	assign1.sh if of
6	xle135	8983	8636	0	08:59	pts/15	00:00:00	ssh ctf.cs.utsarr.net
7	zeh458	9057	1980	0	08:59	pts/7	00:00:00	vim prog.c
8	rslavin	9150	9139	0	08:59	pts/16	00:00:00	ps -af
9	xle135	8636	8628	0	08:58	pts/15	00:00:00	bash

Output:

```
1 User: adz110
2     bash
3     who
4 User: dmq292
5     bash
6     assign1.sh if of
7 User: erg474
8     ls
9 User: xle135
10    bash
11    ssh ctf.cs.utsarr.net
12 User: zeh458
13    vim prog.c
14
15 Earliest Start Time :
16 dmq292    6908    6854    0 Jun04 pts/1    00:00:00 bash
17
18 Latest Start Time :
19 xle135    8983    8636    0 08:59 pts/15    00:00:00 ssh ctf.cs.utsarr.net
```

Also, if there is a tie for earliest or latest start times, take the one with the UID that comes first alphabetically.

Hint: Consider using `sort` to help with grouping.

Script Execution (Part B)

Your program should each be invoked through a single bash file (see below) with input taken from stdin. The resulting output should be printed directly to stdout.

```
$ assign3B.sh < ps.in
or
$ ps -ef | assign3B.sh
```

Assignment Data

Sample input files can be found in:

`/usr/local/courses/rslavin/cs3423/Spring19/assign3.`

Script Files

Your submission should consist of multiple files:

- `assign3A.sh` - a bash script with a single line of code (i.e., one command) for part A
- `assign3B.sh` - a bash script to invoke for part B.
- `assign3B.awk` - the awk program used in `assign3B.awk`

Verifying Your Programs

Part A can be tested with the sample input provided with `passwd.in`.

Part B can be tested with the sample input provided with `ps.in`. Your program should also work with arbitrary input from the `ps -ef` command.

Submission

Turn your assignment in via Blackboard. Your zip file, named `abc123.zip` with your personal `abc123` should contain only your bash and awk files.