System Administration HW3
- Shell Script

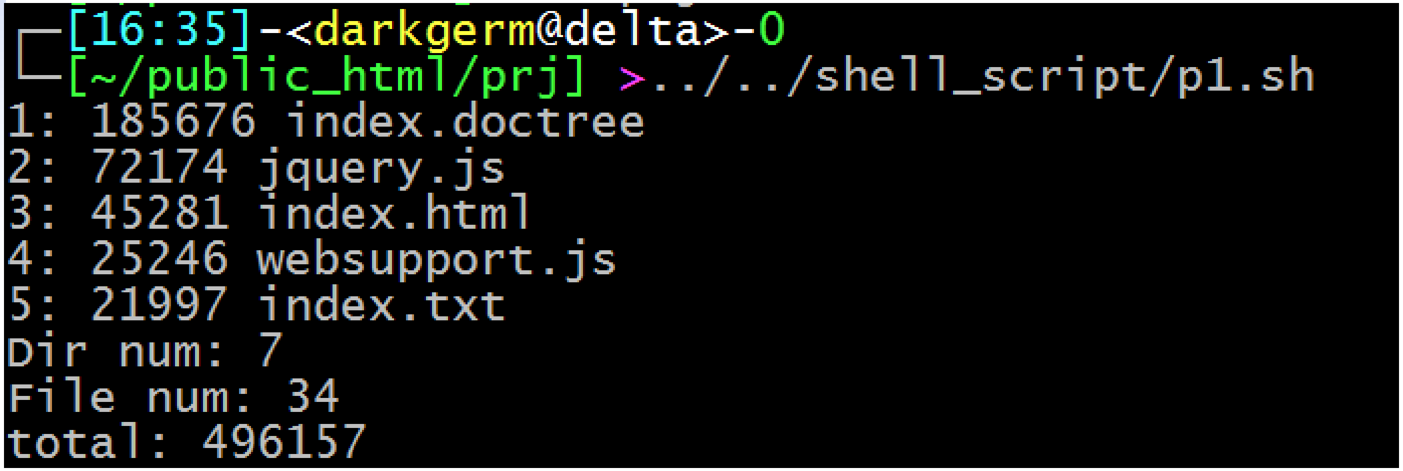
chiachunt

Requirements

- ❑ 3-1: Filesystem Statistic (10%)
  - Use one-line command to show statistic of a directory.
- ❑ 3-2: CPU Usage Plotter (30%)
  - Draw the CPU usage graph by gnuplot.
- ❑ 3-3: System Info Monitor (20+30%)
  - Log the CPU usage with a daemon.
- ❑ Modify code (10%) (Write code by yourself!!)

- ❑ Please write the scripts in Bourne Shell(sh)
  - no score if you use csh or bash or other languages.
- ❑ Due & Demo: 2013/11/28 (Thu.) (After midterm)

3-1: Filesystem Statistic (10%)



```
┌[16:35]-<darkgerm@delta>-0
└[~/public_html/prj] >../../shell_script/p1.sh
1: 185676 index.doctree
2: 72174 jquery.js
3: 45281 index.html
4: 25246 websupport.js
5: 21997 index.txt
Dir num: 7
File num: 34
total: 496157
```

3-1: Filesystem Statistic (10%) - Requirement (1/5)

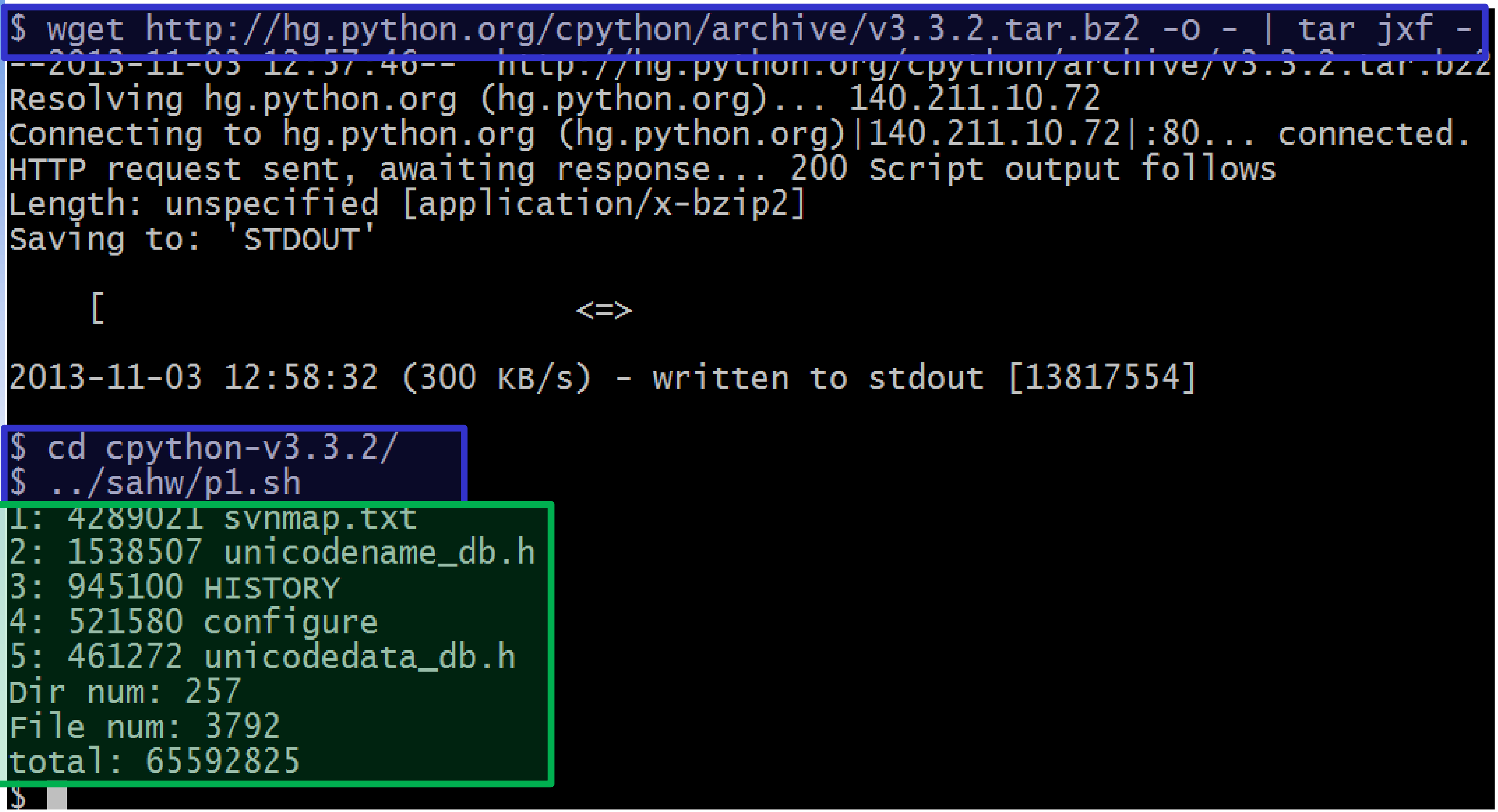
- ❑ Inspect the current directory(“.”) and all sub-directory.
- ❑ Calculate the number of directories.
  - Do not include ‘.’ and ‘..’
- ❑ Calculate the number of files.
- ❑ Calculate the sum of all file size.
- ❑ List the top 5 biggest files.
- ❑ Only consider the regular file. Do not count in the link, FIFO, block device... etc.

3-1: Filesystem Statistic (10%) - Requirement (2/5)

- ❑ Use one-line command.
- ❑ Only pipes (cmd | cmd) and backquotes (`cmd`) are allowed.
- ❑ No temporary files or shell variables.
- ❑ No "&&" "||" ">" ">>" "<" ";" "&"

3-1: Filesystem Statistic (10%) - Requirement (3/5)

- ❑ Sample Test Case



```
$ wget http://hg.python.org/cpython/archive/v3.3.2.tar.bz2 -O - | tar jxf -
--2013-11-03 12:57:46--  http://hg.python.org/cpython/archive/v3.3.2.tar.bz2
Resolving hg.python.org (hg.python.org)... 140.211.10.72
Connecting to hg.python.org (hg.python.org)|140.211.10.72|:80... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [application/x-bzip2]
Saving to: 'STDOUT'

    [                        <=>

2013-11-03 12:58:32 (300 KB/s) - written to stdout [13817554]

$ cd cpython-v3.3.2/
$ ../sahw/p1.sh
1: 4289021 svnmap.txt
2: 1538507 unicodename_db.h
3: 945100 HISTORY
4: 521580 configure
5: 461272 unicodedata_db.h
Dir num: 257
File num: 3792
total: 65592825
$
```

3-1: Filesystem Statistic (10%) - Requirement (4/5)

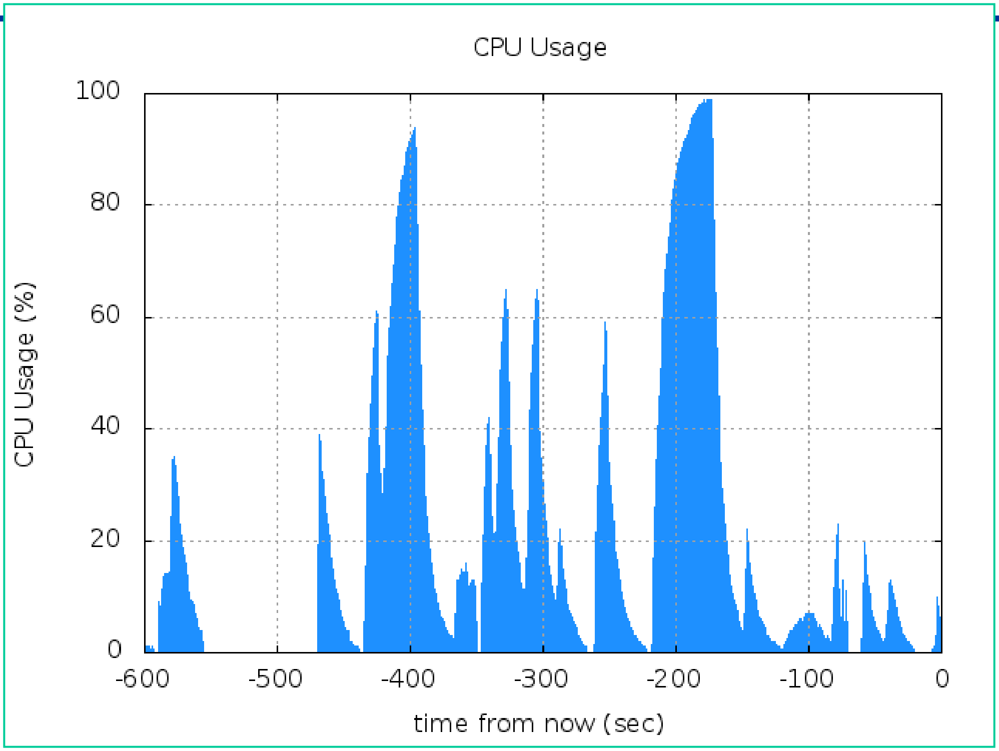
- ❑ Hint
  - ls(1) with argument A and R

3-1: Filesystem Statistic (10%) - Requirement (5/5)

❑ Grade

- File is executable. (2%)
- List top 5 file size and name. (2%)
- Dir num is correct. (2%)
- File num is correct. (2%)
- Total size is correct. (2%)

3-2: CPU Usage Plotter (30%)



3-2: CPU Usage Plotter (30%) - Requirement (1/7)

- ❑ Use gnuplot to draw the CPU usage.
- ❑ CPU usage is logged in a log file. You only need to read and plot it.
- ❑ The output file is in png format. To open it, you can:
  - start X11 and open it with any picture viewer.
  - download it via FTP.
  - execute "python –m SimpleHTTPServer" and browse http://your.ip:8000/

3-2: CPU Usage Plotter (30%) - Requirement (2/7)

- ❑ cpuplot [-h] [-o out_file_name] [-t type] [-c color] -n <60-600>
- ❑ -h print the help.
- ❑ -o set the output file name. (default: out.png)
- ❑ -t set the graph type. (one of ‘filledcurve’, ‘lines’. default: ‘filledcurve’)
- ❑ -c set graph color. (in hexadecimal form, default: #1E90FF)
- ❑ -n set the number of point should use. (must be set. should be in range [60-600])
- ❑ Read LOGFILE environment variable. If it is not set, use “/tmp/sysmonitor”

3-2: CPU Usage Plotter (30%) - Requirement (3/7)

- ❑ If type is not one of ‘filledcurve’, ‘lines’, you should print error message and help.
  - type should be one of 'filledcurve' and 'lines'.
- ❑ If color is not in hexadecimal form (a leading sharp ‘#’ and 6 hex digits.), you should print error message and help.
  - color format error.
- ❑ If the number specified by –n is not in [60-600], you should print error message and help.
  - num should be in range 60 - 600.
- ❑ It doesn’t matter which is checked first.
- ❑ You can generate temporary files, but you need to clean them when exit.

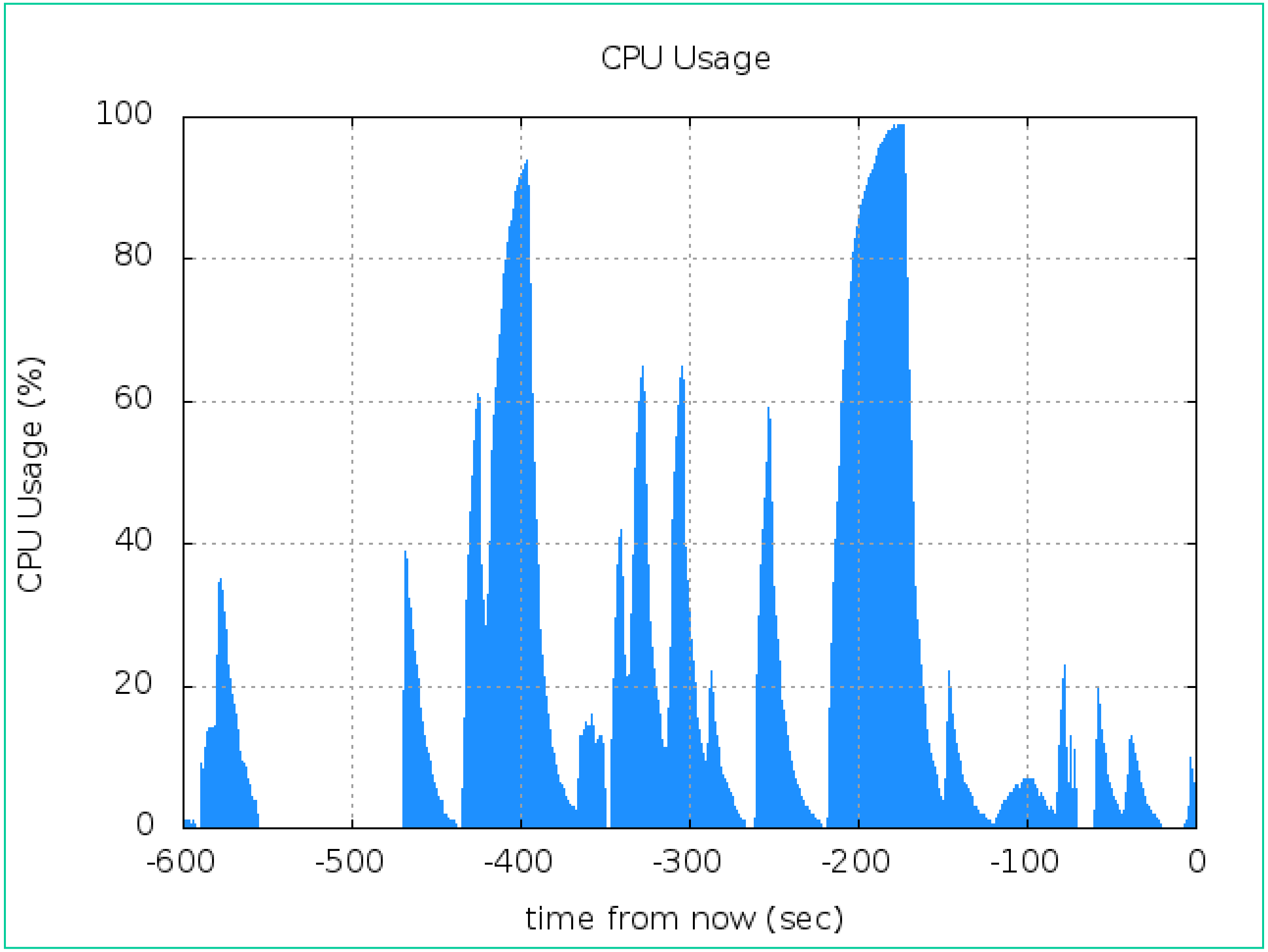
3-2: CPU Usage Plotter (30%) - Requirement (4/7)

❑ Image output format

- The title should be “CPU Usage”.
- The y-title should be “CPU Usage (%)”.
- The x-title should be ”time from now (sec)”.
- The y-axis range should be 0-100.
- The x-axis means “time from now”. You can’t predict future, so the value should be negative.
- The CPU usage is appended to the log file every second. If you get “-n 60” it means you should draw the last 60 lines in the log file.

3-2: CPU Usage Plotter (30%) - Requirement (5/7)

- ❑ Sample image output.



3-2: CPU Usage Plotter (30%) - Requirement (6/7)

❑ Hint

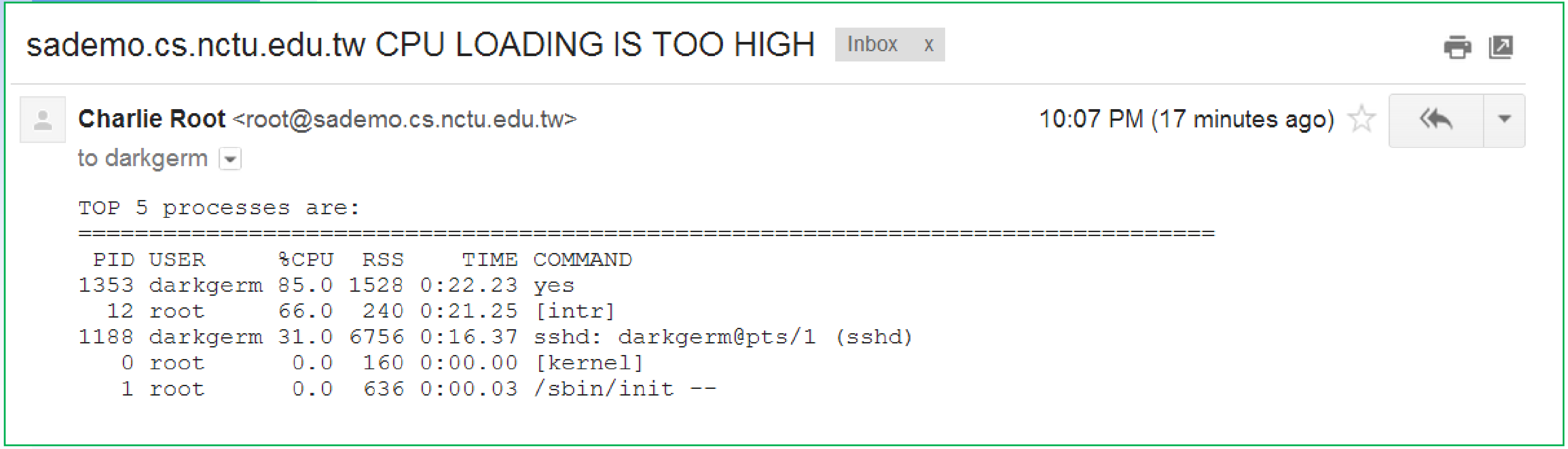
- /usr/ports/math/gnuplot
- gnuplot(1)
- getopt(1)

3-2: CPU Usage Plotter (30%) - Requirement (7/7)

❑ Grade

- Parse the arguments correctly. (10%)
- Fool-proof. (4%)
- Draw the graph correctly. (10%)
- Read the right log file. (3%)
- Get the right data. (3%)

3-3: System Info Monitor (50%)



sademo.cs.nctu.edu.tw CPU LOADING IS TOO HIGH

Charlie Root <root@sademo.cs.nctu.edu.tw>

to darkgerm

10:07 PM (17 minutes ago)

```
TOP 5 processes are:
=================================================================================
 PID USER      %CPU  RSS    TIME COMMAND
1353 darkgerm 85.0 1528 0:22.23 yes
  12 root     66.0  240 0:21.25 [intr]
1188 darkgerm 31.0 6756 0:16.37 sshd: darkgerm@pts/1 (sshd)
   0 root      0.0  160 0:00.00 [kernel]
   1 root      0.0  636 0:00.03 /sbin/init --
```

3-3: System Info Monitor (50%) - Requirement (1/8)

- **Part 1: Get and calculate the CPU usage and log it to a file.**
- Your script should be in /usr/local/bin/.
- sysmonitor --logfile filename --email yourname@server
- Log the usage per second to the file specify by --logfile
- Do not need to implement fool-proof. You can add additional arguments if you need.
- You should check the user who run this script. If the user is not root, you should print some error message and exit.

3-3: System Info Monitor (50%) - Requirement (2/8)

- ❑ **Part 1: Get and calculate the CPU usage and log it to a file. (cont'd)**
- ❑ Get CPU usage of each process from ps(1).
- ❑ (sum of CPU usage exclude [idle] process) / (sum of CPU usage)
- ❑ Assume the PID of idle process is 11.
- ❑ You can only call ps once per second.

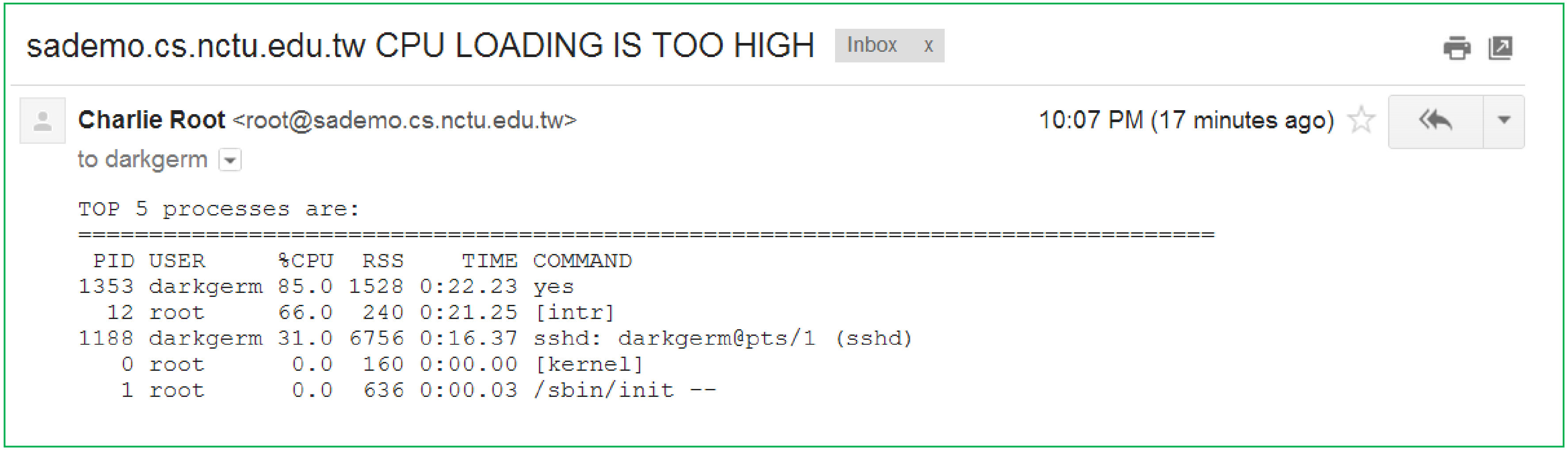
3-3: System Info Monitor (50%) - Requirement (3/8)

- ❑ **Part 1: Get and calculate the CPU usage and log it to a file. (cont’d)**
- ❑ If the CPU usage is higher than 90% over 5 seconds, you should send a email out to notify you.
- ❑ When you send a mail, you are allowed to call ps at the second time.
- ❑ Only send one mail if the loading is continuously high.

3-3: System Info Monitor (50%) - Requirement (5/8)

❑ Sample mail format

- You should exclude the [idle] process.
- List top 5 processes of CPU usage. You should sort the result by CPU usage.
- You should display the same columns as below.



sademo.cs.nctu.edu.tw CPU LOADING IS TOO HIGH

Charlie Root <root@sademo.cs.nctu.edu.tw> 10:07 PM (17 minutes ago)

to darkgerm

```
TOP 5 processes are:
================================================================================
 PID USER     %CPU  RSS    TIME COMMAND
1353 darkgerm 85.0 1528 0:22.23 yes
  12 root     66.0  240 0:21.25 [intr]
1188 darkgerm 31.0 6756 0:16.37 sshd: darkgerm@pts/1 (sshd)
   0 root      0.0  160 0:00.00 [kernel]
   1 root      0.0  636 0:00.03 /sbin/init --
```

3-3: System Info Monitor (50%) - Requirement (4/8)

- ❑ **Part 2: Write a RC script to manage it.**
- ❑ Your RC script should be in /usr/local/etc/rc.d/.
- ❑ Start it when system start.
- ❑ Can use service(8) to start or stop it.
- ❑ You should log the running pid in "/var/run/sysmonitord.pid"

3-3: System Info Monitor (50%) - Requirement (6/8)

❑ Hint

- /etc/rc.subr
- rc.subr(8)
- http://www.freebsd.org/doc/en_US.ISO8859-1/articles/rc-scripting/index.html
- rcorder(8)
- mail(1)

3-3: System Info Monitor (50%) - Requirement (7/8)

❑ Hint – Use sh to write a daemon.

- You can take the following code as reference.
- If you execute it without argument, it will fork a process with –a argument and exit.
- The process with –a argument will run something it should run. (ex: sleep 999999)

```
#!/bin/sh

echo "pid $$"
if [ -z $1 ]; then
        echo "no args, pid $$"
        $0 -a &
        echo "good bye $$"
        exit
else
        echo "has arg, pid $$"
        sleep 999999
fi
```

3-3: System Info Monitor (50%) - Requirement (8/8)

❑ Grade - Part 1

- Can parse the argument. (4%)
- The CPU usage is correct. (3%)
- Can send the mail at the right time. (3%)
- Can send the mail and format is correct. (10%)

❑ Grade - Part 2

- Can start on system start. (10%)
- Can use service(8) to start it. (10%)
- Can use service(8) to stop it. (10%)

Make a clean environment

- ❑ env -i PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin sh

Help

- IRC channel #nctuNASA on freenode
- Newsgroup cs.course.sysadmin
- BBS bs2.to board CS-SysAdmin
- Email ta@nasa.cs.nctu.edu.tw
- Goto CSCC to ask professional 3F!