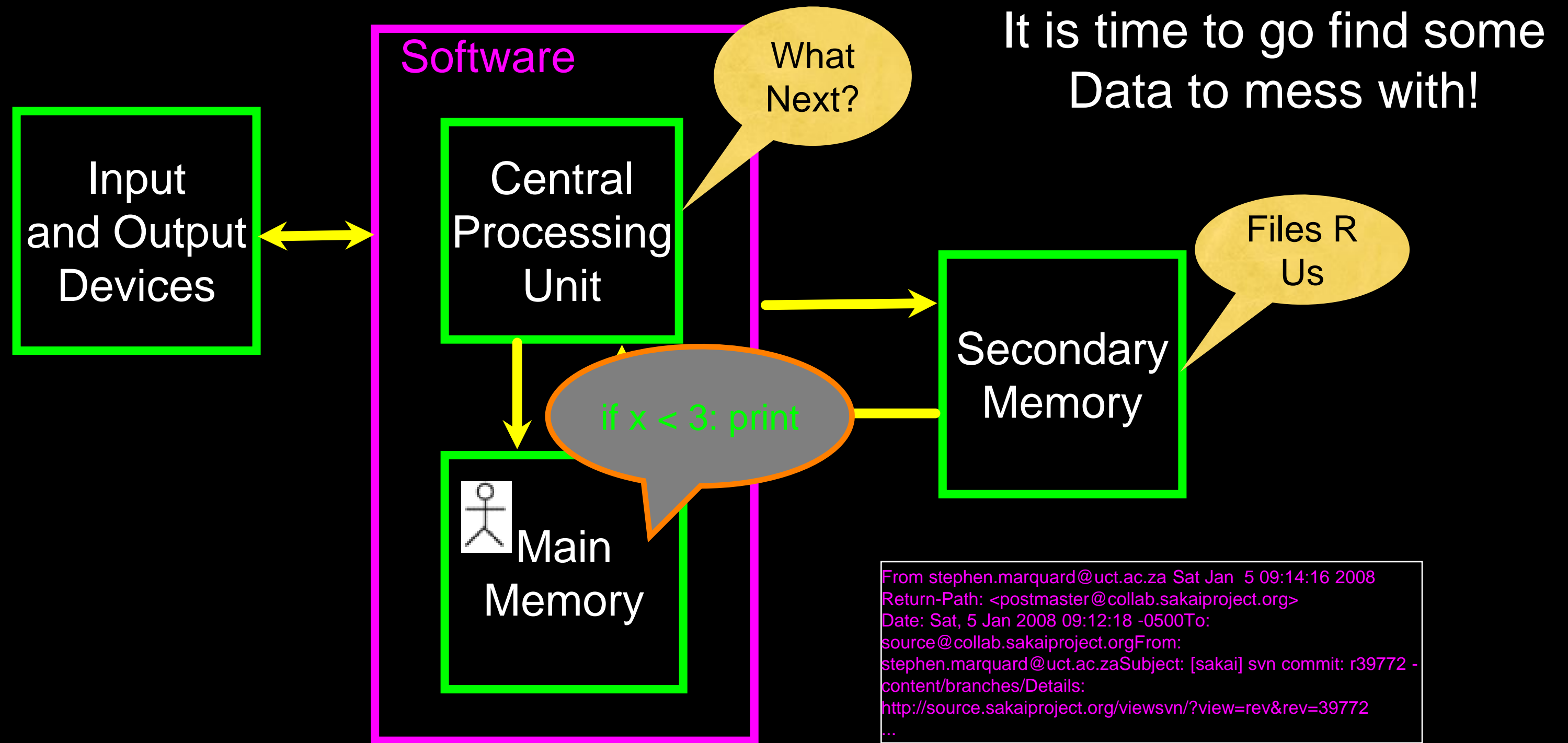


# Reading Files

## Chapter 7

Python for Everybody  
[www.py4e.com](http://www.py4e.com)





# File Processing

A text file can be thought of as a sequence of lines

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

Return-Path: <postmaster@collab.sakaiproject.org>

Date: Sat, 5 Jan 2008 09:12:18 -0500

To: source@collab.sakaiproject.org

From: stephen.marquard@uct.ac.za

Subject: [sakai] svn commit: r39772 - content/branches/

Details: <http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772>

<http://www.py4e.com/code/mbox-short.txt>

# Opening a File

- Before we can read the contents of the file, we must tell Python which file we are going to work with and what we will be doing with the file
- This is done with the `open()` function
- `open()` returns a “file handle” - a variable used to perform operations on the file
- Similar to “File -> Open” in a Word Processor

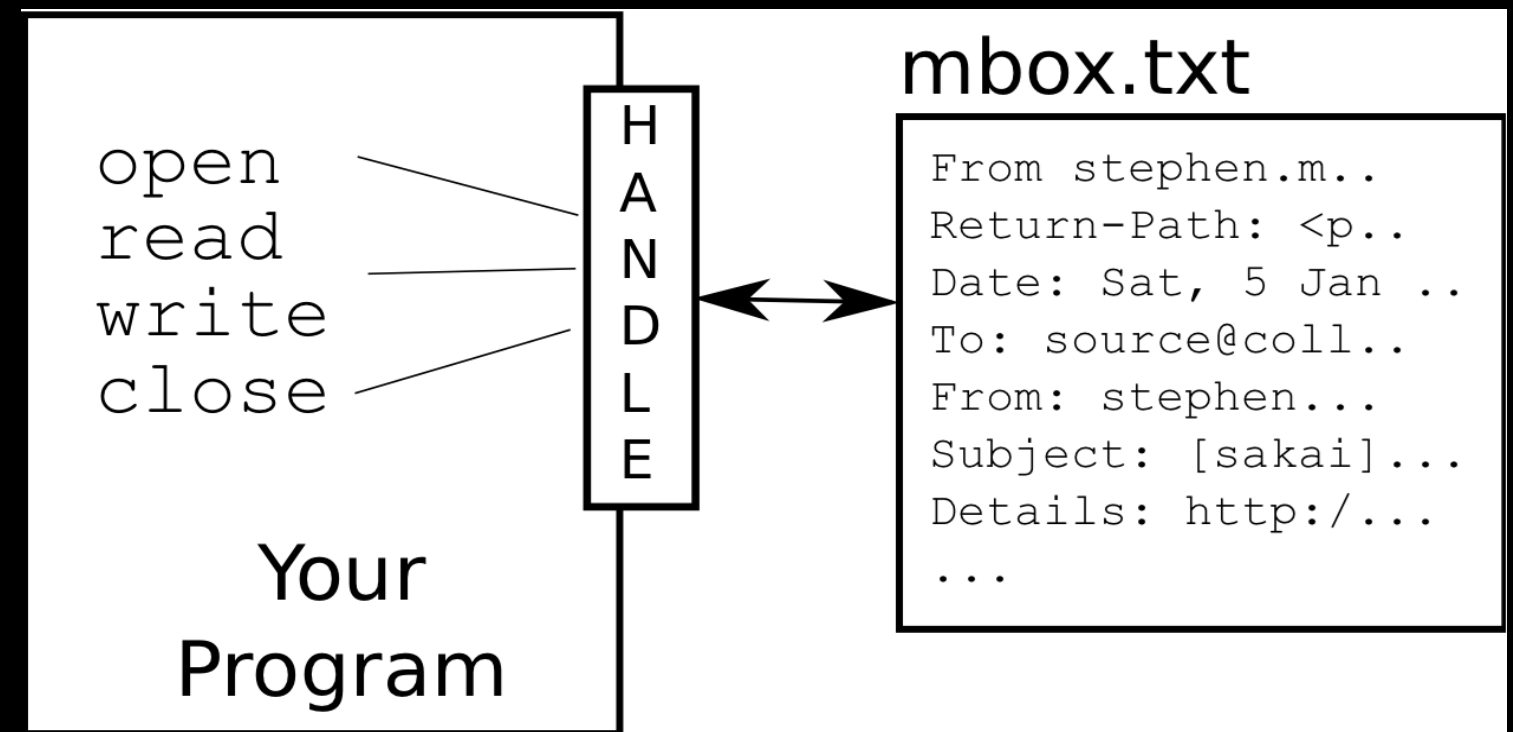
# Using open()

```
fhand = open('mbox.txt', 'r')
```

- `handle = open(filename, mode)`
- returns a handle use to manipulate the file
- filename is a string
- mode is optional and should be 'r' if we are planning to read the file and 'w' if we are going to write to the file

# What is a Handle?

```
>>> fhand = open('mbox.txt')
>>> print(fhand)
<_io.TextIOWrapper name='mbox.txt' mode='r' encoding='UTF-8'>
```



# When Files are Missing

```
>>> fhand = open('stuff.txt')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or
directory: 'stuff.txt'
```

# The `newline` Character

- We use a special character called the “`newline`” to indicate when a line ends
- We represent it as `\n` in strings
- `Newline` is still one character - not two

```
>>> stuff = 'Hello\nWorld!'
>>> stuff
'Hello\nWorld!'
>>> print(stuff)
Hello
World!
>>> stuff = 'X\nY'
>>> print(stuff)
X
Y
>>> len(stuff)
3
```



# File Processing

A text file can be thought of as a sequence of lines

```
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008
Return-Path: <postmaster@collab.sakaiproject.org>
Date: Sat, 5 Jan 2008 09:12:18 -0500
To: source@collab.sakaiproject.org
From: stephen.marquard@uct.ac.za
Subject: [sakai] svn commit: r39772 - content/branches/
```

```
Details: http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772
```

# File Processing

A text file has **newlines** at the end of each line

```
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008\nReturn-Path: <postmaster@collab.sakaiproject.org>\nDate: Sat, 5 Jan 2008 09:12:18 -0500\nTo: source@collab.sakaiproject.org\nFrom: stephen.marquard@uct.ac.za\nSubject: [sakai] svn commit: r39772 - content/branches/\n\nDetails: http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772\n
```

# Reading Files in Python

# File Handle as a Sequence

- A **file handle** open for read can be treated as a **sequence** of strings where each line in the file is a string in the sequence
- We can use the **for** statement to iterate through a **sequence**
- Remember - a **sequence** is an ordered set

```
xfile = open('mbox.txt')  
for cheese in xfile:  
    print(cheese)
```

# Counting Lines in a File

- Open a **file** read-only
- Use a **for** loop to read each line
- **Count** the lines and print out the number of lines

```
fhand = open('mbox.txt')  
count = 0  
for line in fhand:  
    count = count + 1  
print('Line Count:', count)
```

```
$ python open.py  
Line Count: 132045
```

# Reading the \*Whole\* File

We can **read** the whole file (newlines and all) into a **single string**

```
>>> fhand = open('mbox-short.txt')
>>> inp = fhand.read()
>>> print(len(inp))
94626
>>> print(inp[:20])
From stephen.marquar
```

# Searching Through a File

We can put an **if** statement in our **for** loop to only print lines that meet some criteria

```
fhand = open('mbox-short.txt')
for line in fhand:
    if line.startswith('From: ') :
        print(line)
```

# OOPS!

What are all these blank  
lines doing here?

From: `stephen.marquard@uct.ac.za`

From: `louis@media.berkeley.edu`

From: `zqian@umich.edu`

From: `rjlowe@iupui.edu`

...



# OOPS!

What are all these blank lines doing here?

- Each line from the file has a **newline** at the end
- The **print** statement adds a **newline** to each line

```
From: stephen.marquard@uct.ac.za\n\nFrom: louis@media.berkeley.edu\n\nFrom: zqian@umich.edu\n\nFrom: rjlowe@iupui.edu\n\n...
```

# Searching Through a File (fixed)

- We can strip the whitespace from the right-hand side of the string using `rstrip()` from the string library
- The newline is considered “white space” and is **stripped**

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if line.startswith('From:') :
        print(line)
```

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
....
```

# Skipping with continue

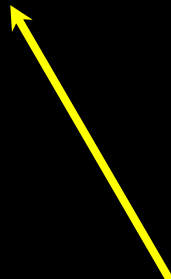
We can conveniently skip a line by using the `continue` statement

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if not line.startswith('From: ') :
        continue
    print(line)
```

# Using `in` to Select Lines

We can look for a string anywhere `in` a `line` as our selection criteria

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if not '@uct.ac.za' in line :
        continue
    print(line)
```



```
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008
X-Authentication-Warning: set sender to stephen.marquard@uct.ac.za using -f
From: stephen.marquard@uct.ac.za
Author: stephen.marquard@uct.ac.za
From david.horwitz@uct.ac.za Fri Jan  4 07:02:32 2008
X-Authentication-Warning: set sender to david.horwitz@uct.ac.za using -f...
```

```
fname = input('Enter the file name: ')
fhand = open(fname)
count = 0
for line in fhand:
    if line.startswith('Subject:'):
        count = count + 1
print('There were', count, 'subject lines in', fname)
```

# Prompt for File Name

Enter the file name: mbox.txt

There were 1797 subject lines in mbox.txt

Enter the file name: mbox-short.txt

There were 27 subject lines in mbox-short.txt

# Bad File Names

```
fname = input('Enter the file name: ')
try:
    fhand = open(fname)
except:
    print('File cannot be opened:', fname)
    quit()

count = 0
for line in fhand:
    if line.startswith('Subject:') :
        count = count + 1
print('There were', count, 'subject lines in', fname)
```

Enter the file name: mbox.txt

There were 1797 subject lines in mbox.txt

Enter the file name: na na boo boo

File cannot be opened: na na boo boo

# Summary

- Secondary storage
- Opening a file - file handle
- File structure - newline character
- Reading a file line by line with a for loop
- Searching for lines
- Reading file names
- Dealing with bad files



# Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) of the University of Michigan School of Information and [open.umich.edu](http://open.umich.edu) and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

... Insert new Contributors and Translators here