Intermediate C Programming

Introduction

Sep 27th 2013

Instructor Introduction

- Dr. Zhou Su

- Doctor obtained from Waseda University
- Assistant Professor, Waseda University
- zhousu@asagi.waseda.jp

TA Introduction

- Feel free to ask any help during the course!

Attendance Requirement

- TA will assign a roll card to take attendance at the beginning of the class.
- For the students who are late more than 10 minutes can not attend the class.
- PLEASE log in the Unix system during the time of class to ensure your attendance.
- The status of your attendance can be confirmed by course navi.

Report

- All students need to submit two reports.
- The details of reports will be given during the time of exercise class.

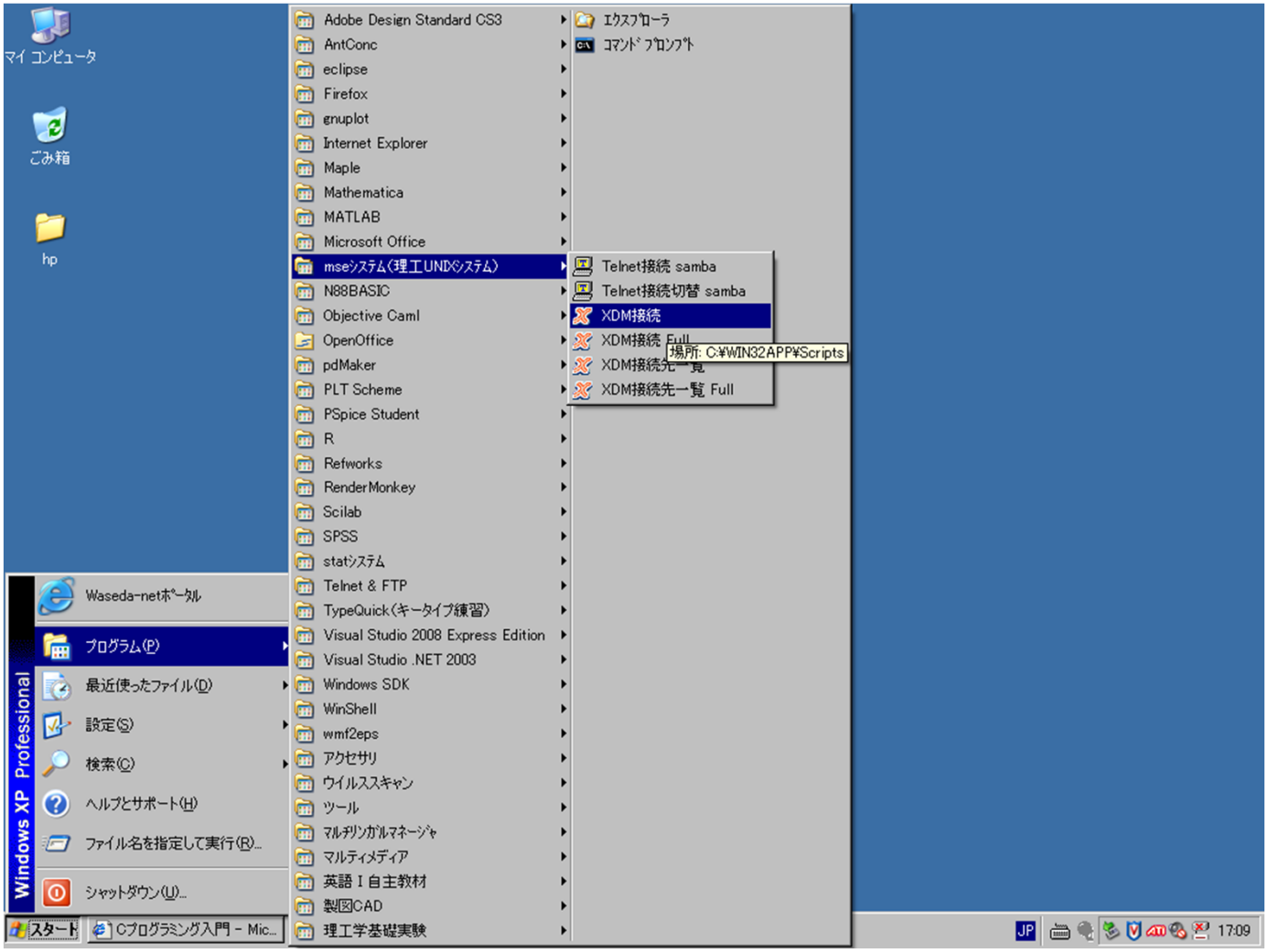
Grade Policy

- Grades will be computed by the reports and the percentage of attendance
- There is a final test for this course.
- The percentage of attendance must be more then 2/3 to get the credit.

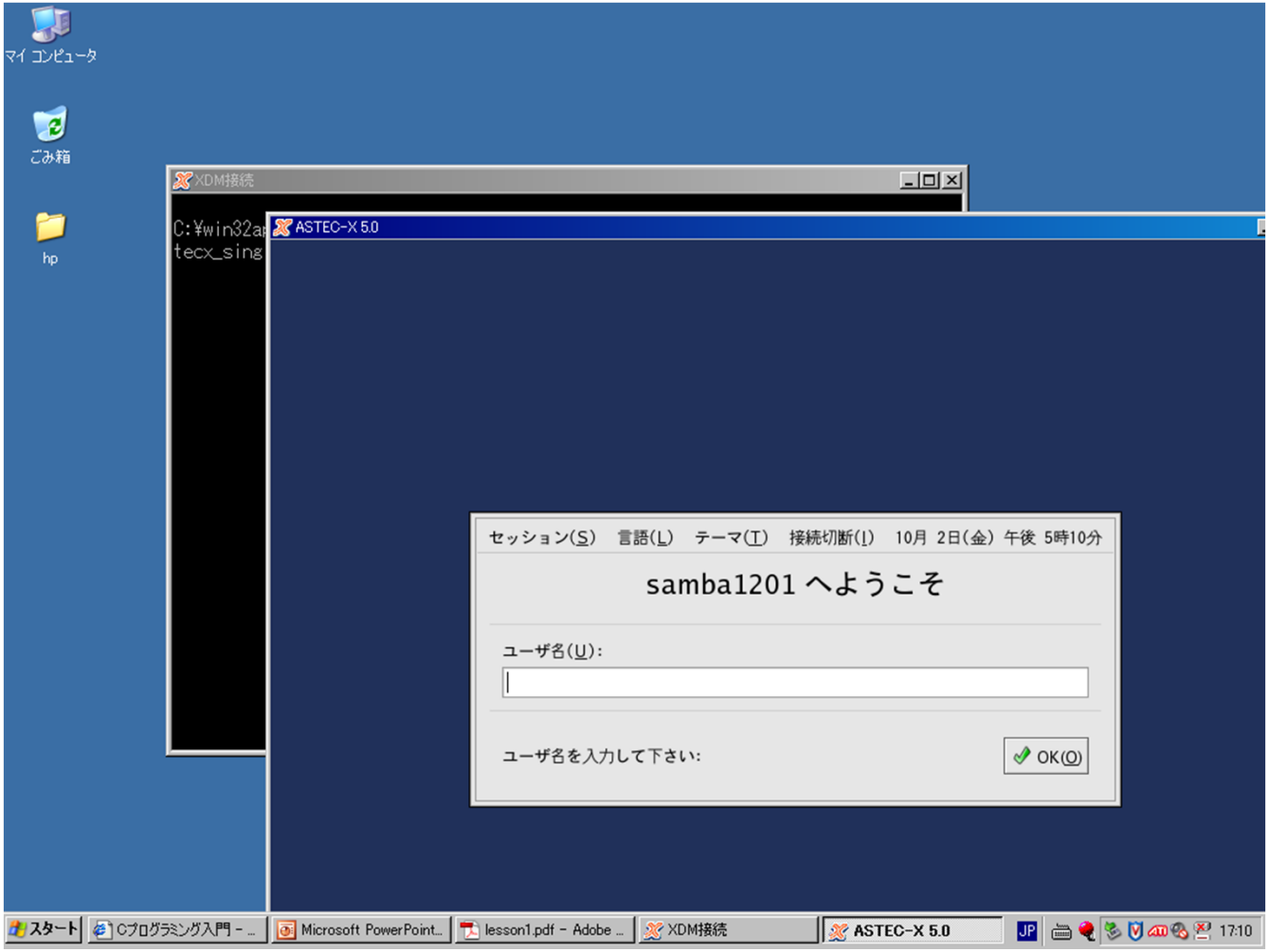
Today's Outline

- OS
- Log-in and Log-out in Unix System
- Review of C Programming

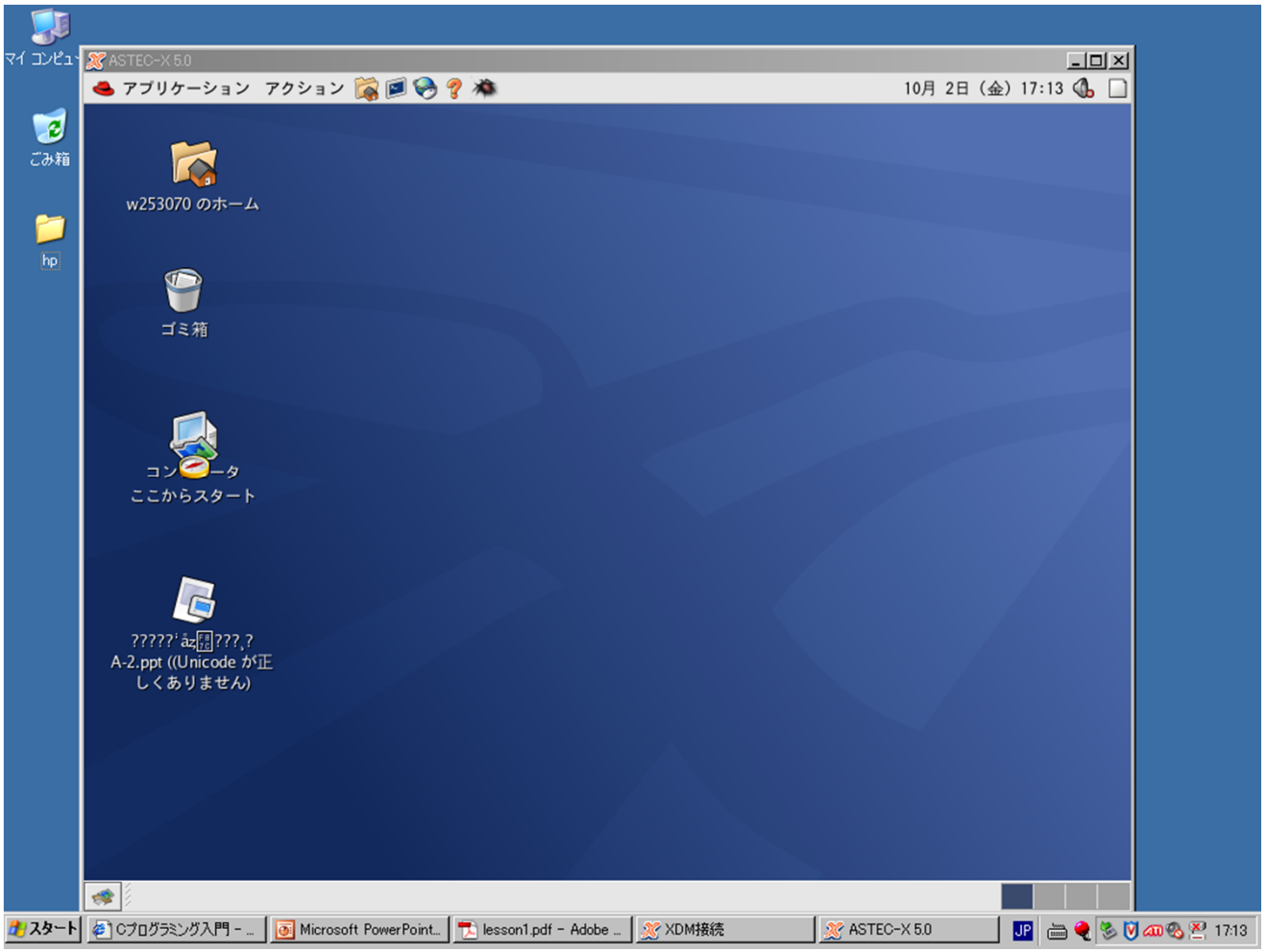
Log-in and Log-out in Unix System



Log-in and Log-out in Unix System



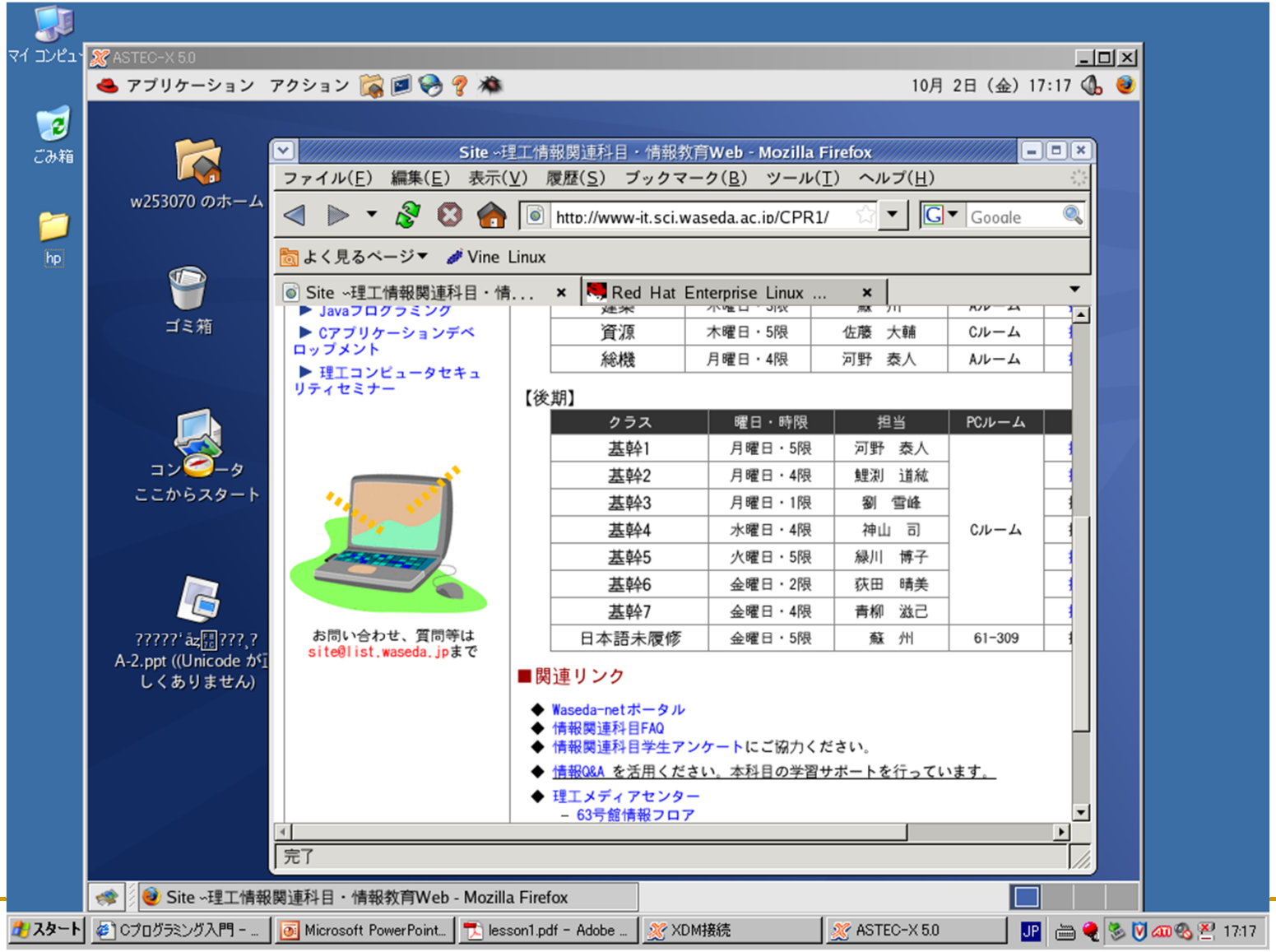
GNOME


ASTEC-X 5.0
アプリケーション　アクション
10月 2日（金）17:13
w253070 のホーム
ゴミ箱
ここからスタート
A-2.ppt ((Unicode が正
しくありません)
スタート
Microsoft PowerPoint...
lesson1.pdf - Adobe ...
XDM接続
ASTEC-X 5.0
JP
17:13

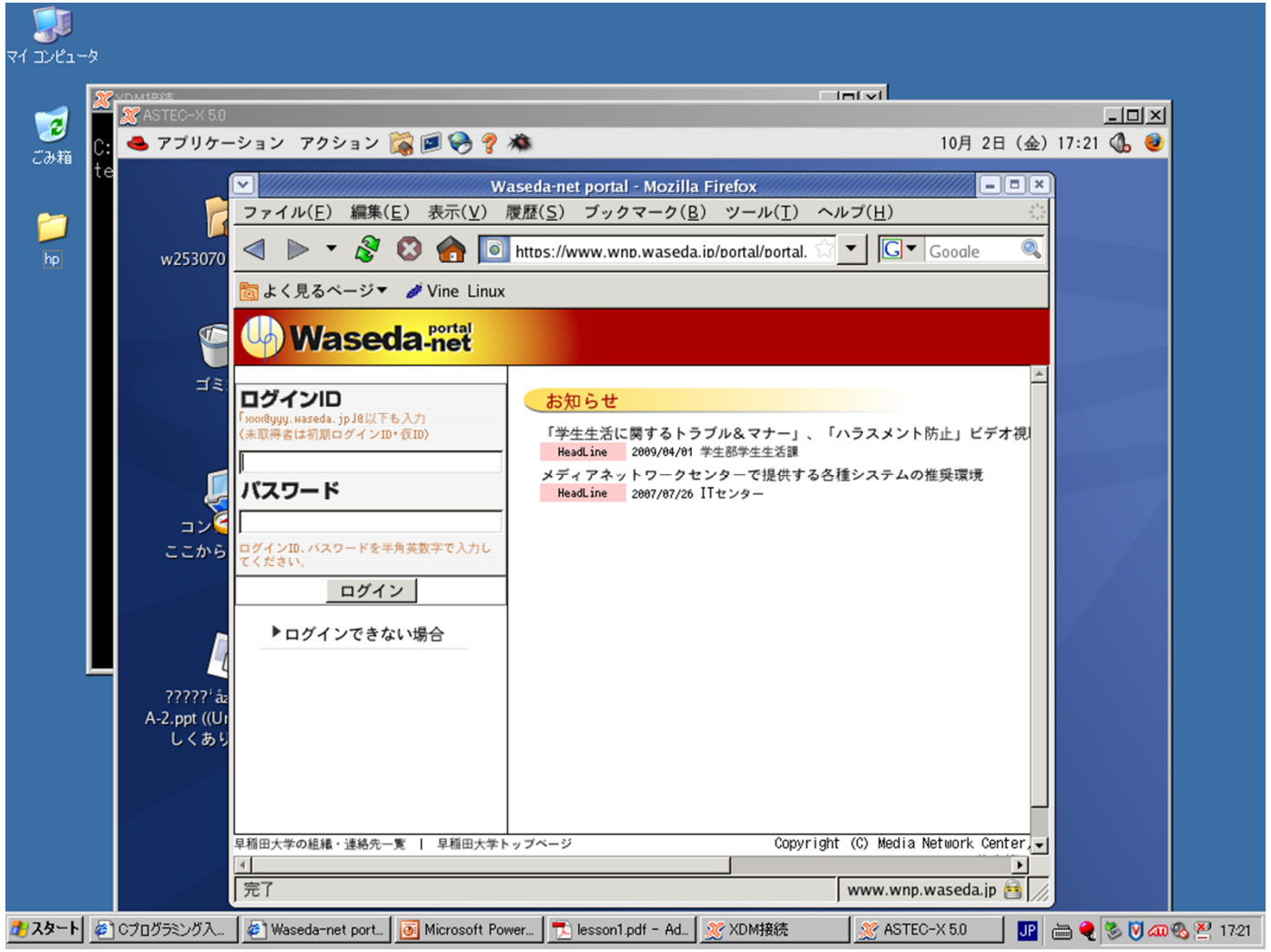
Web Browser


ASTEC-X 5.0
アプリケーション
アクション
10月 2日 (金) 17:14
Wolfram Mathematica
アクセサリ
インターネット
オフィス
グラフィックス
サウンドとビデオ
システムツール
システム設定
プログラミング
個人設定
ネットワーク・サーバ
ファイル・ブラウザ
ヘルプ
Firefox Web Browser
IRC
Konqueror
Thunderbird Email
インターネット・メッセンジャー
ビデオ会議
電子メール
redhat.com | - Mozilla Firefox
ブックマーク(B)
ツール(T)
ヘルプ(H)
Red Hat Enterprise Linux ...
Downloads | Fedora | Red Hat Network
Developers
Training
Support
Company
Welcome to Mozilla Firefox
The accompanying version of Mozilla Firefox utilizes website information services ("Services"), such as
features, which are provided by the Mozilla Corporation and made available to you under additional terms
Services, you consent to the terms of the referenced Mozilla Firefox Website Services Agreement. If you
these terms, do not use the Services and disable the Services in Edit > Preferences > Security and
options for both: "Tell me if the site I'm visiting is a suspected attack site" and "Tell me if the site I'm vis
suspected forgery".
完了
A-2.ppt ((Unicode が正しくありません)
redhat.com | - Mozilla Firefox
スタート
Cプログラミング入門 - ...
Microsoft PowerPoint...
lesson1.pdf - Adobe ...
XDM接続
ASTEC-X 5.0
JP
17:14

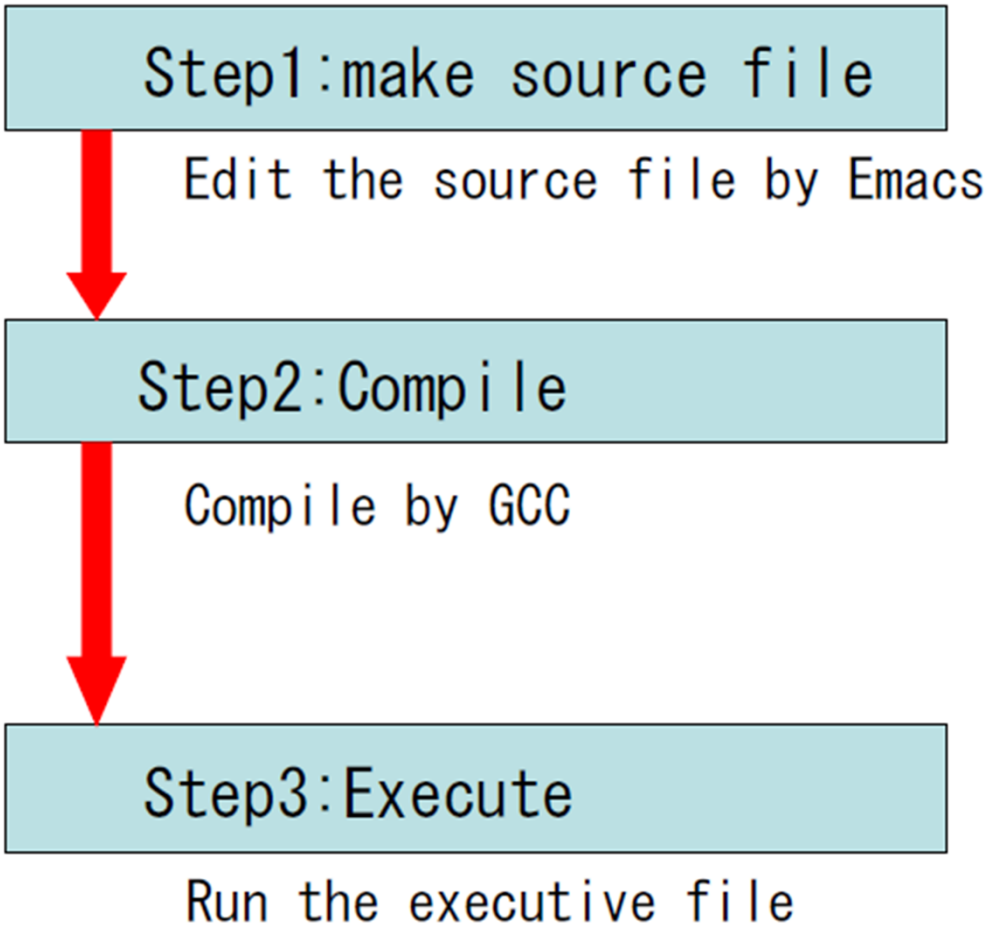
Course HP


ASTEC-X 5.0
アプリケーション アクション
10月 2日 (金) 17:17
w253070 のホーム
ごみ箱
hp
ゴミ箱
ここからスタート
Site ~理工情報関連科目・情報教育Web - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
http://www-it.sci.waseda.ac.jp/CPR1/
よく見るページ Vine Linux
Site ~理工情報関連科目・情... Red Hat Enterprise Linux ...
▶ Javaプログラミング
▶ Cアプリケーションデベロップメント
▶ 理工コンピュータセキュリティセミナー
お問い合わせ、質問等は site@list.waseda.jpまで
資源 木曜日・5限 佐藤 大輔 Cルーム
総機 月曜日・4限 河野 泰人 Aルーム
【後期】
クラス 曜日・時限 担当 PCルーム
基幹1 月曜日・5限 河野 泰人
基幹2 月曜日・4限 鯉渕 道紘
基幹3 月曜日・1限 劉 雪峰
基幹4 水曜日・4限 神山 司 Cルーム
基幹5 火曜日・5限 緑川 博子
基幹6 金曜日・2限 荻田 晴美
基幹7 金曜日・4限 青柳 滋己
日本語未履修 金曜日・5限 蘇 州 61-309
■関連リンク
◆ Waseda-netポータル
◆ 情報関連科目FAQ
◆ 情報関連科目学生アンケートにご協力ください。
◆ 情報Q&A を活用ください。本科目の学習サポートを行っています。
◆ 理工メディアセンター
– 63号館情報フロア
完了
Site ~理工情報関連科目・情報教育Web - Mozilla Firefox
スタート
Cプログラミング入門 - ...
Microsoft PowerPoint...
lesson1.pdf - Adobe ...
XDM接続
ASTEC-X 5.0
JP
17:17

Course Navi


マイ コンピュータ
ごみ箱
hp
ASTEC-X 5.0
アプリケーション アクション
10月 2日 (金) 17:21
w253070
Waseda-net portal - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
https://www.wnp.waseda.jp/portal/portal.
Google
よく見るページ
Vine Linux
Waseda-net portal
ログインID
「xxx@yyy.waseda.jp」@以下も入力
〈未取得者は初期ログインID・仮ID〉
パスワード
ログインID、パスワードを半角英数字で入力してください。
ログイン
▶ログインできない場合
お知らせ
「学生生活に関するトラブル＆マナー」、「ハラスメント防止」ビデオ視
HeadLine 2009/04/01 学生部学生生活課
メディアネットワークセンターで提供する各種システムの推奨環境
HeadLine 2007/07/26 ITセンター
早稲田大学の組織・連絡先一覧 | 早稲田大学トップページ
Copyright (C) Media Network Center
完了
www.wnp.waseda.jp
スタート
Cプログラミング入...
Waseda-net port...
Microsoft Power...
lesson1.pdf - Ad...
XDM接続
ASTEC-X 5.0
JP
17:21

C Programming



Emacs

Make New File

```
emacs  hello.c  &
```

```
#include <stdio.h>

int main(void)
{
    printf("hello, world¥n");

    return 0;
}
```

¥ could be ＼(back slash) according the comupter environment

⇒ Click "Save" icon to save the file

Compile

GCC(GNU Compiler Collection)
GNOME：

```
gcc [Option] [File]
```

```
$ ls ↵
hello.c
$ gcc  hello.c ↵          (Compile「hello.c」)
$
```

⇒If there is no error message, compiling is success
  Otherwise, grammar mistakes exist

⇒Revising the source file by Emacs
  And compile it again

Execute

(a.out) will be created after the compiling

Run a.out at GNOME :

```
$ ls ↵
a.out  hello.c
$ ./a.out ↵                    (Run「a.out」)
hello, world
```

Variables

- Variables are used to declare values and strings
- Declaration is neccesary before using variables.
- Type of Variables
- Type can be divided by :

Type	Details
int	Integer
long	Long integer
float	Floating point numbers
double	Double precision
char	Character

- Variables Declaration :

  Type Name;

For Example
```
int  height, weight;
double  height_m;
```

Output

printf( ) : To output variable values and charactors

charactors

```
printf("hello, world¥n");
```

- contents within "...." will be printed
- change line「¥n」.

variable values

```
printf("%d¥n", height);
```

- Output variable「height」' value.
- %: specifier
- define the type
  and the interpretation of the value

specifier	detials
%d	decimal integer
%x	hexadecimal integer
%f	Decimal floating point
%e	exponent
%c	character

Input from the keyboard

scanf( ) : input the value of variable from the keyboard

```
scanf("%d", &height);
```

- The value inputted by the keyboard will be assinged to 「height」
- 「&」 before the variables

specifier	detials
%d	decimal integer
%f	Decimal floating point
%lf	Decimal double floating point
%c	character

Four rules of arithmetic

- Arithmetic Operators

Operation	Operator
addition	+
subtraction	-
multiplication	*
division	/
modulus	%

(Integer division truncates any fractional part)

(Example. 10 % 3 → 1)

- Precedence: + - -> * / ->parenthesis

  例) 1 + 2 * 3 → 7

  (1 + 2) * 3 → 9

- the calculation between integer and real number is looked on as the calculation among integers

  例) 175 / 100 → 1 (Integer division truncates any fractional part)

  175 / 100.0 → 1.75 (real number(175)dicided by real number(100.0))

If-Else

if ～ else

```
if ( expression) {
    statement1;
    statement2;
    ...
    statementm;
} else {
    statementm+1;
    statementm+2;
    ...
    statementn;
}
```

if the expression is true;
statement1, ... statementm
will be executed, otherwise,
statementm+1, ...statementn
will be executed.

•the part after「else」can be omitted

While

- Loop- While

```
while ( expression) {
    statement1;
    statement2;
    ...
    statementn;
}
```

If the expression is true,

statement1, statement2, … statementn

will be executed.

The deatils of expression
are avalaible in the 4th class

for

- a method to define a loop as while.

```
for ( initialization_expression;loop_condition;increment_expression) {
    statement1;
    statement2;
     …
    statementn;
}
```

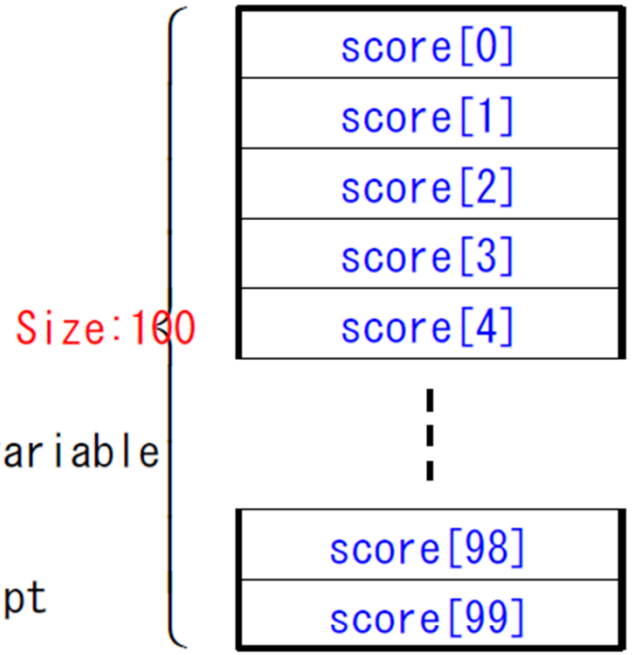
- initialization_expression is executed before execution of the loop starts.
- The execution of the loop continues until the loop_condition is false.
- The increment_expression is executed at the end of each loop iteration.

Array

- `int score[100];`

  type — number of size

- score[0]...score[99]: a block of 100 consecutive elements can be used.
  (score[100]is not avalaible)



- Each element can be used as an int- variable
- The number within []is called subscript
- The subscript can be used as a variable

```
int score[100];

score[10] = 75;
```

=

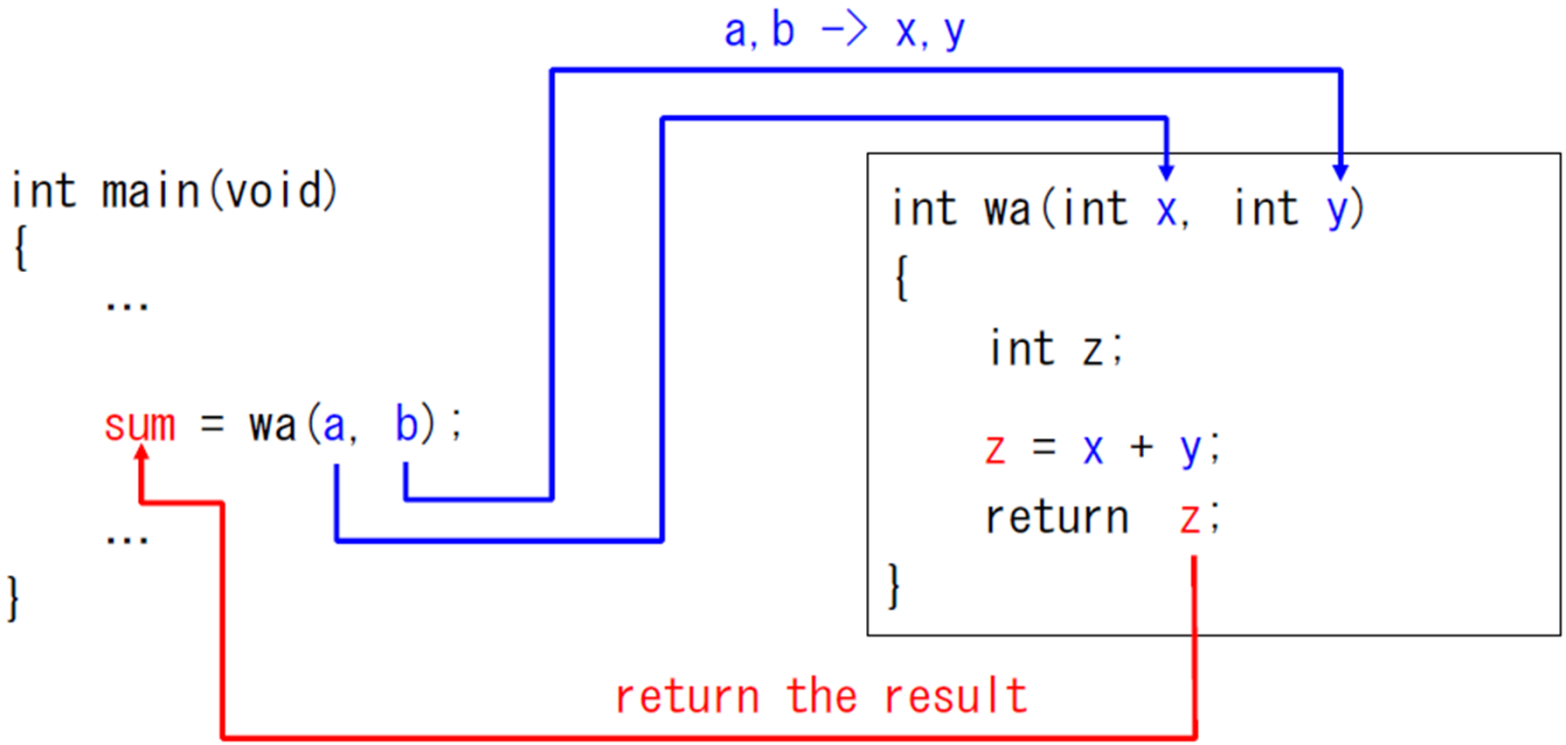
```
int score[100];
int count;

count = 10;
score[count] = 75;
```

Functions Definition

when sample.cis executed

- In C programming, the execution of program is started from main()



Pointers

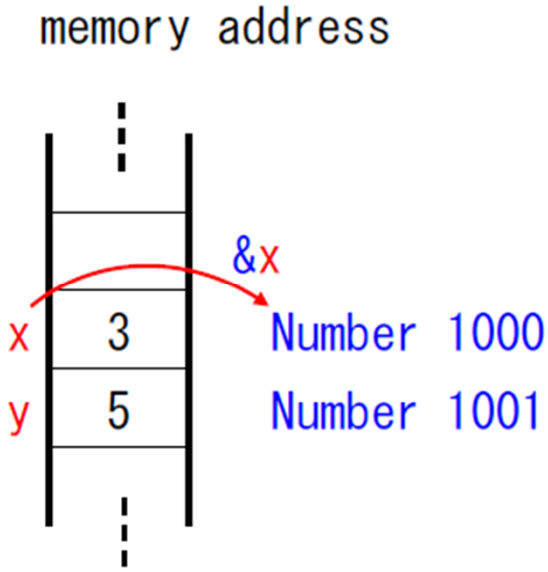
To get the address of variables ....

For examples,

```
int x;
```

by writing &x
we can get the address of variable X.

- this address is called 「x's pointer」.
- & is called address operator



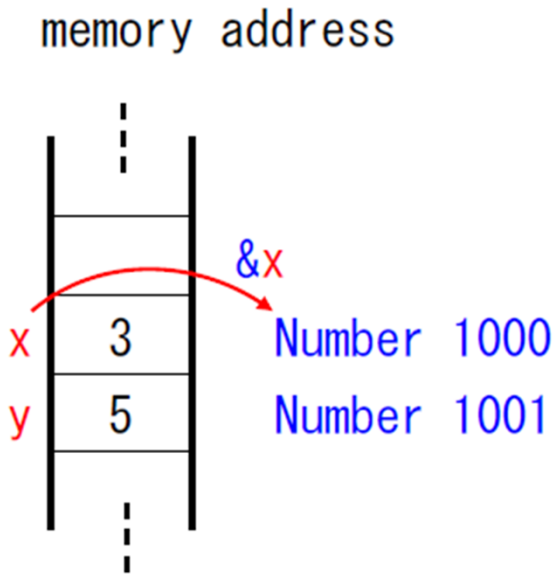
Pointers

To get the address of variables ....

- To declare a pointer

```
int *x_p;
```

x_p will be a pointer to show
the address of a variable (int type)



- In this case,

```
x_p = &x;
```

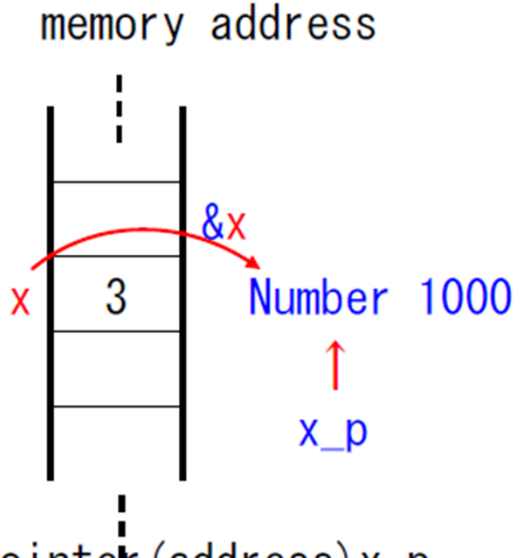
← x_p stores the address of x

means that pointer x_p will store the address of x
⇒that is to say: 「x_p points to x」

Pointers

- the image of the following operation can be shown in the right figure

```
int x = 3;
int *x_p;
x_p = &x;
```



- by using * operator, we can get the value of variable which is stored in pointer (address) x-p

- * is called indirection operator.

pointer3.c

```
#include <stdio.h>

int main(void)
{
    int a = 123, b;
    int *p;

    p = &a;
    b = *p;
    printf("a = %d, b = %d, *p = %d¥n", a, b, *p);

    a = 200;
    b = 300;
    printf("a = %d, b = %d, *p = %d¥n", a, b, *p);

    p = &b;
    printf("a = %d, b = %d, *p = %d¥n", a, b, *p);

    return 0;
}
```