

# エディタの使い方

中村 匡秀

計算科学演習I 第1回講義資料

※本資料のオリジナルは関和弘先生によって作成されました。

# 今日やること

- Emacsの起動方法
- 基本キー操作
- ファイル操作
- バッファ
- Emacsの終了方法

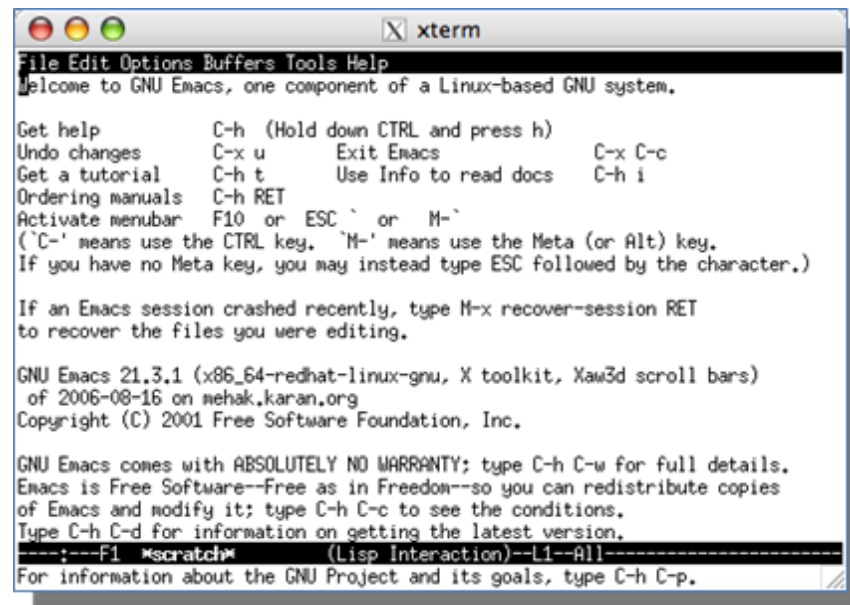
# キー操作の表記法

- C-□
  - CTRL を押しながら...
- M-□
  - ALT を押しながら...
  - うまく行かない時は, ESCを押ししてから□を押す

# Emacsの起動方法

- ターミナルで

```
$ ssh scalar
$ emacs
```



```
xterm
File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of a Linux-based GNU system.

Get help          C-h (Hold down CTRL and press h)
Undo changes     C-x u      Exit Emacs          C-x C-c
Get a tutorial    C-h t      Use Info to read docs  C-h i
Ordering manuals C-h RET
Activate menubar F10 or ESC ` or M-`
(`C-' means use the CTRL key. `M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)

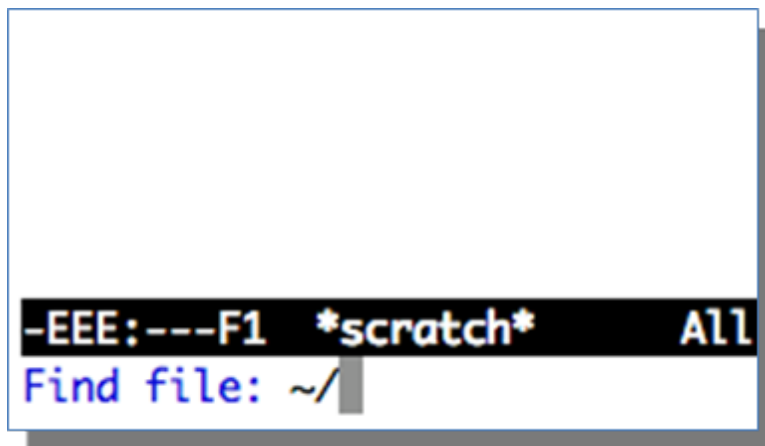
If an Emacs session crashed recently, type M-x recover-session RET
to recover the files you were editing.

GNU Emacs 21.3.1 (x86_64-redhat-linux-gnu, X toolkit, Xaw3d scroll bars)
of 2006-08-16 on mehak.karan.org
Copyright (C) 2001 Free Software Foundation, Inc.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.
---:--F1 *scratch* (Lisp Interaction)--L1--All-----
For information about the GNU Project and its goals, type C-h C-p.
```

# ファイルの作り方(開き方)

- control キーを押しながら, x, f を続けて押す  
– **C-x C-f**
- エコー領域(画面一番下)に「Find file: ~/」というメッセージが出る
- ファイル名(test.c)を入力して return



```
-EEE:---F1 *scratch* All  
Find file: ~/
```

タブを押すとファイル名などの補完が出来る

# キー操作取り消し

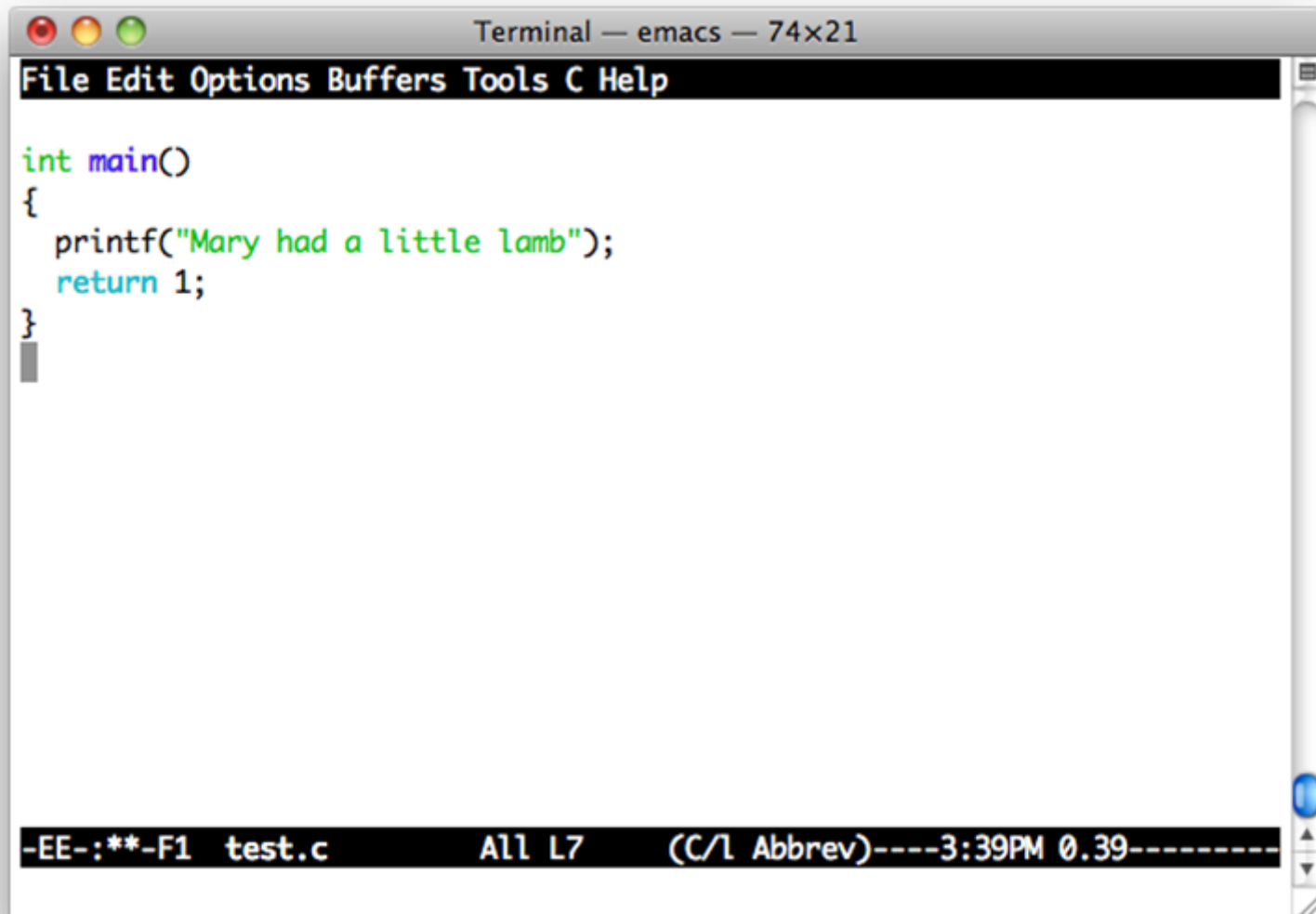
- わけが分からなくなったら

– C-g

で直前のキー操作をキャンセル(編集のundoとは違う)

エコー領域によく分からない表示が出ている時は、消えるまで CTRL+g を押し続ければ良い

# ファイルの編集




The image shows a screenshot of an Emacs terminal window. The title bar reads "Terminal — emacs — 74x21". The menu bar contains "File Edit Options Buffers Tools C Help". The main editing area displays the following C code:

```
int main()
{
    printf("Mary had a little lamb");
    return 1;
}
```

The status bar at the bottom shows: "-EE-:\*\*-F1 test.c All L7 (C/l Abbrev)----3:39PM 0.39-----".

# ファイルの保存

- C-x C-s



The image shows a screenshot of the Emacs status bar. The status bar is divided into two sections. The top section has a black background with white text: "-EEE:---F1 test.c All". The bottom section has a white background with black text: "Wrote /Users/seki/test.c".



# シェル／Emacsに戻る

- **C-z** で emacs を**中断**し, コンパイル, 実行

```
$ gcc test.c
```

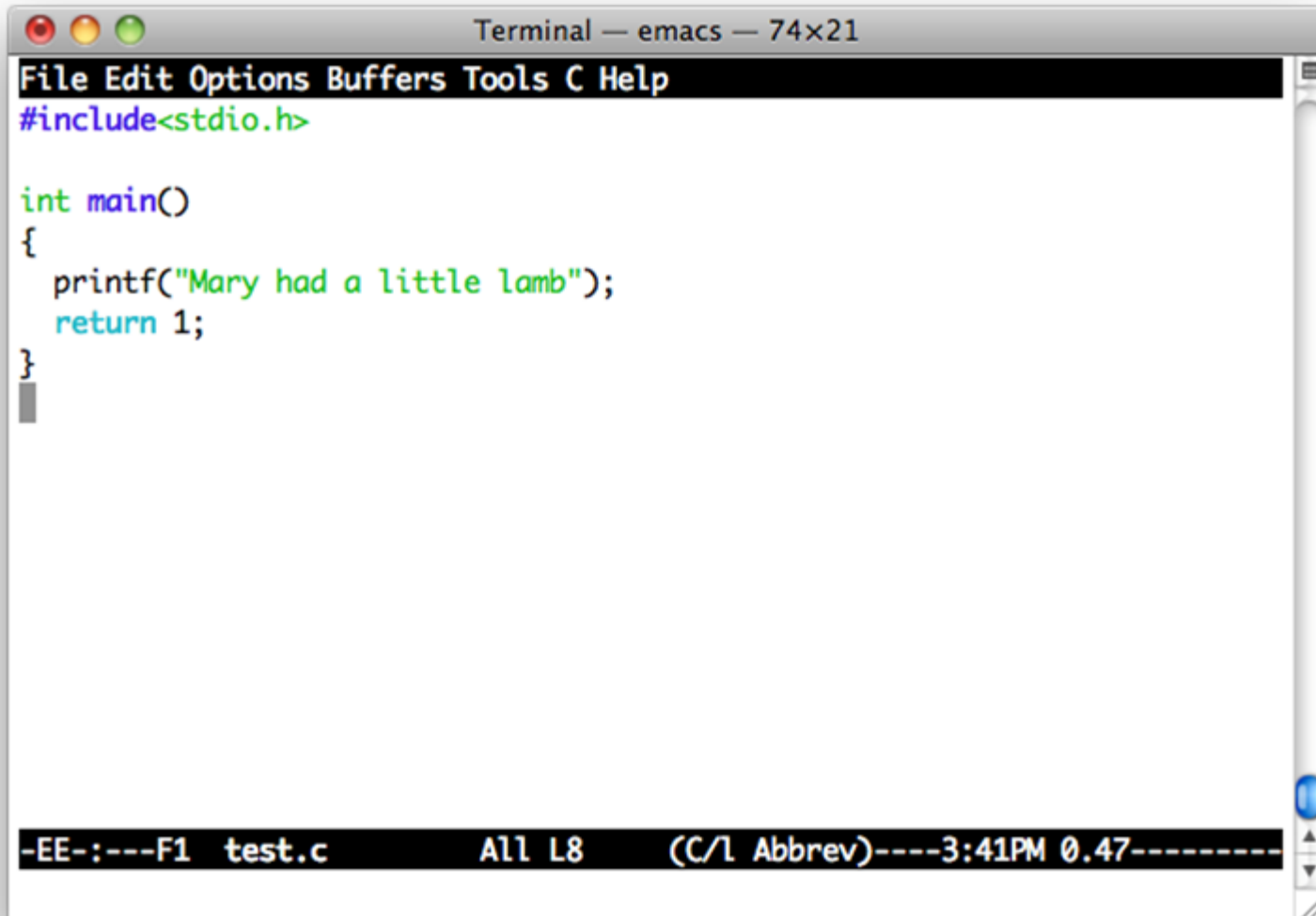
```
test.c: In function === main ===:
```

```
...
```

```
$ fg ← 中断しているプログラム (emacs) を再開
```

毎回 Emacs を終了する必要はない！

# ファイルの編集



The image shows a screenshot of an Emacs terminal window. The window title is "Terminal — emacs — 74x21". The menu bar includes "File Edit Options Buffers Tools C Help". The main content area displays the following C code:

```
#include<stdio.h>

int main()
{
    printf("Mary had a little lamb");
    return 1;
}
```

The status bar at the bottom shows: "-EE-:---F1 test.c All L8 (C/l Abbrev)----3:41PM 0.47-----".

# コピー & ペースト

- 手順

1. **C-@** (C-space). コピー(カット)領域開始位置を指定
2. コピー(カット)領域終了位置までカーソルを移動
3. **M-w** (ESC を押してからw). 上の1で指定した開始位置から現在のカーソル位置までをコピー
4. 貼付けたいところまでカーソルを移動
5. **C-y**. コピー(カット)した領域をペースト

# 編集の取り消し(undo)

- **C-\_** (つまりSHIFT+CTRL+¥)
  - 直前の編集を元に戻す
  - 続けて押せば, さらに前の編集を取り消せる

# 練習

- コピー&ペーストで

Mary had a little lamb,  
Little lamb, little lamb.  
Mary had a little lamb.

と表示されるようにせよ。(入力しても簡単だが、コピー&ペーストすること)

# カット & ペースト

- 手順

1. **C-@** (C-space). コピー(カット)領域開始位置を指定
2. コピー(カット)領域終了位置までカーソルを移動
3. **C-w**. 上の1で指定した位置から現在のカーソル位置までをカット
4. 貼付けたいところまでカーソルを移動
5. **C-y**. コピー(カット)した領域をペースト

# 便利なカット

- **C-k**. 現在のカーソル位置から行末までをカット
  - 一行まるまるカット & ペーストする時に便利
- そのままカットを繰り返すと、連続した複数行をまとめてクリップボードに入れられる
  - 複数行まるまるカット & ペーストする時に便利

行末の改行までカットするためには、  
カットを2回連続して行う

# 練習

- C-w によるカット, C-k によるカットをそれぞれ試してみなさい。(もちろん, その後ペーストしてみる)



# 便利なカーソル移動

- **C-v**. 1画面下に移動(進む)
- **M-v**(ESCを押してからv). 1画面上に移動(戻る)
- **C-e**. 行末へ移動
- **C-a**. 行頭へ移動


# 検索によるカーソル移動

- **C-s.** 入力文字を検索, 移動(カーソル位置から下へ)
  - 探している文字列を見つけたら, 「enter」やカーソルキーで編集モードに戻る
  - 文字列を見つけたあとに CTRL+s をもう一度押すと, その次の出現箇所を検索
- **C-r.** 同じく検索, 移動(カーソル位置から上へ)

# 練習

- 次のファイルを開き, 色々なカーソル移動方法を試してみよ

/tmp/week1/songs.txt

A terminal window showing a file completion prompt. The prompt is "----:%%-F1 \*Completions\* (Completion". Below it, the text "Find file: /tmp/week1/songs.txt" is displayed, with a cursor at the end of the path. The terminal has a dark background and light-colored text.

```
----:%%-F1 *Completions* (Completion
Find file: /tmp/week1/songs.txt
```

# バッファ

- Emacsでのファイルの作成・編集は、**バッファ**と呼ばれる作業領域で行われる
- 通常、画面に表示されているバッファは一つだが、内部的には**複数のバッファを保持**できる



バッファを切り替えることで、異なるバッファ  
(ファイル)間でもコピー&ペーストが可能

# バッファの切り替え

- **C-x b** (C-x を押してから b). エコー領域で切り替えたいファイル名を指定

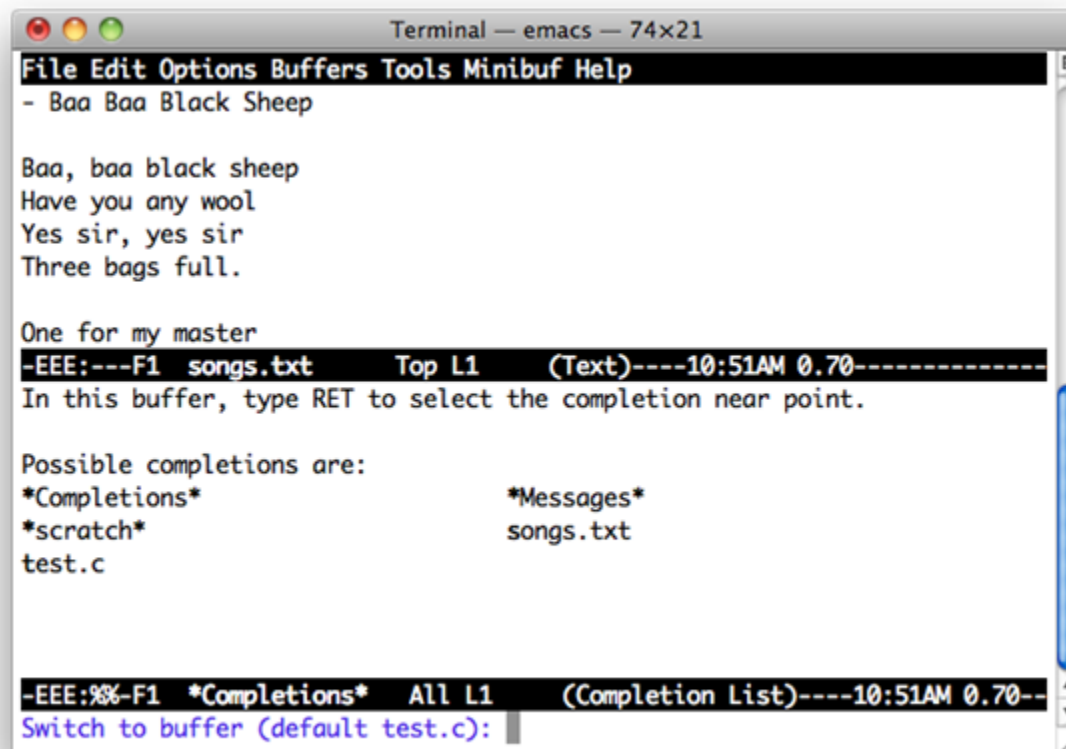
```
- Hey Diddle Diddle

Hey diddle diddle,
The cat and the fiddle,
The cow jumped over the moon,
The little dog laughed to see such sport,
-EEE:---F1 songs.txt      Top L1      (Text)----10:48A
Switch to buffer (default test.c):
```

※ そのままEnterを押すと、デフォルトのバッファに切り替わる

# バッファの切り替え

- 続けてタブを押すと、画面が二分割されて、切り替え可能なバッファ名が表示される



```
Terminal — emacs — 74x21
File Edit Options Buffers Tools Minibuf Help
- Baa Baa Black Sheep

Baa, baa black sheep
Have you any wool
Yes sir, yes sir
Three bags full.

One for my master
-EEE:---F1 songs.txt Top L1 (Text)----10:51AM 0.70-----
In this buffer, type RET to select the completion near point.

Possible completions are:
*Completions*          *Messages*
*scratch*              songs.txt
test.c

-EEE:%%-F1 *Completions* All L1 (Completion List)----10:51AM 0.70--
Switch to buffer (default test.c):
```

# 複数バッファの表示

- **C-x 2.** 画面が2分割される.
- **C-x o.** バッファ間でカーソルを移動.
- **C-x 0.** 今カーソルがあるバッファだけを残して,  
画面分割解除

# 練習

- バッファを切り替えて, songs.txt から適当な歌詞を test.c の printf 部分に貼付けなさい.
- 複数バッファの表示を試してみなさい. 試したあとは, 画面分割を解除して, 元の一バッファ表示に戻すこと.



# Emacs の終了の仕方

- **C-x C-c**

# 課題(練習)

- 自宅,あるいは研究室のパソコンから scalar にログインしなさい.
  - OSは問わない
- 自宅,あるいは研究室のパソコンから scalar にファイルをコピーしなさい.
  - 何のファイルでも良い
- scalar 上で, 下記を実行しなさい(**締切:次回授業日の前日23:59まで**)
  - \$ echo | mail -s **自分のアカウント名** masa-n