

## システムプログラム第3回レポート

2024年10月18日  
学籍番号：35714121  
名前：福富隆大

① 講義資料内のe3-1m.c, e3-1a-ia32.s 2つのファイルを作成してコンパイルし、実行ファイルを作成して、これを実行しその結果を報告してください。

### 実行結果

エラーのためなし

### 作成したファイル内容、結果について

作成したファイルをコンパイルするとエラーとなった。

原因を調べるためにエラーを調べてみると書き方が合っていないようだった。

cseでarchコマンドを用いてアーキテクチャを確認してみるとx86\_64と表示されIA32とは違うことがわかった。よって、このファイルがエラーとなった原因はアーキテクチャによって命令形式と動作が違うからである。

② e3-1a-ia32.sを修正して、popした値をedx, eaxの順に格納するようにして、①と同様に実行ファイルを作成してこれを実行し、その結果を報告してください。

### 実行結果

エラーのためなし

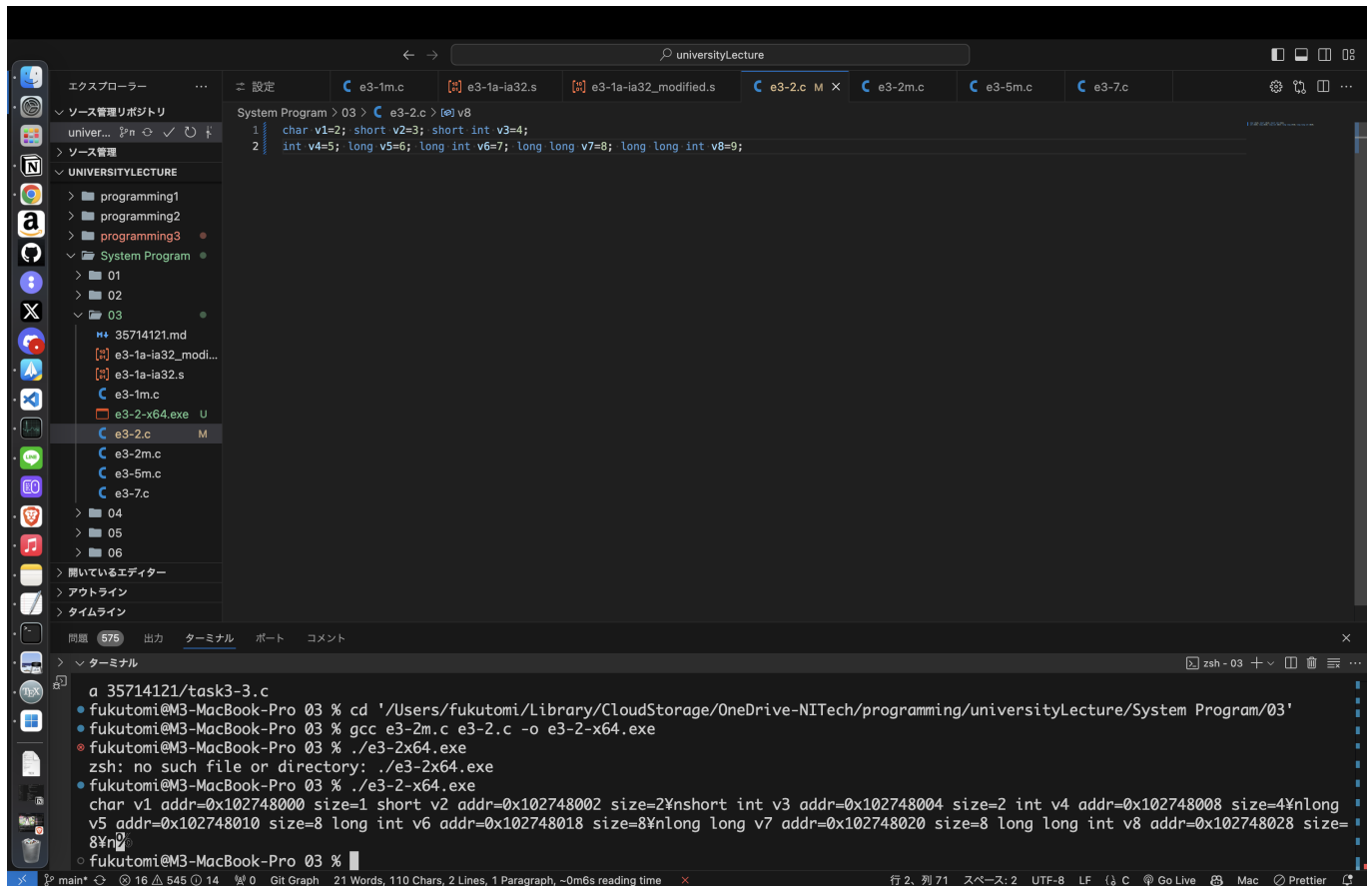
### 作成したファイル内容、結果について

①と同じ理由の理由でエラーになった。popした値をedx, eaxの順に格納するには、もともとpop edx, pop eaxの順に実行しているので、それをpop edx, pop eaxのように順番を変えよう。こうすることでpopした値を格納する順番がedx→eaxとなる。

③ e3-2.cの変数に入れる値を変更したものとe3-2m.cのファイルを作成してコンパイルし、実行ファイルを実行した結果を報告してください。

### 実行結果

画像の下部分のターミナルに実行結果あります



The screenshot shows a code editor with a sidebar on the left displaying a file tree. The main editor area shows a C program with two lines of code: `1 char v1=2; short v2=3; short int v3=4;` and `2 int v4=5; long v5=6; long int v6=7; long long v7=8; long long int v8=9;`. Below the code, a terminal window displays the execution output. The output shows the memory addresses and sizes of the variables: `char v1 addr=0x102748000 size=1 short v2 addr=0x102748002 size=2 int v3 addr=0x102748004 size=4 long v4 addr=0x102748008 size=8 long int v5 addr=0x102748010 size=8 long long v6 addr=0x102748018 size=16 long long int v7 addr=0x102748020 size=16 long long int v8 addr=0x102748028 size=16`.

## 作成したファイル内容について

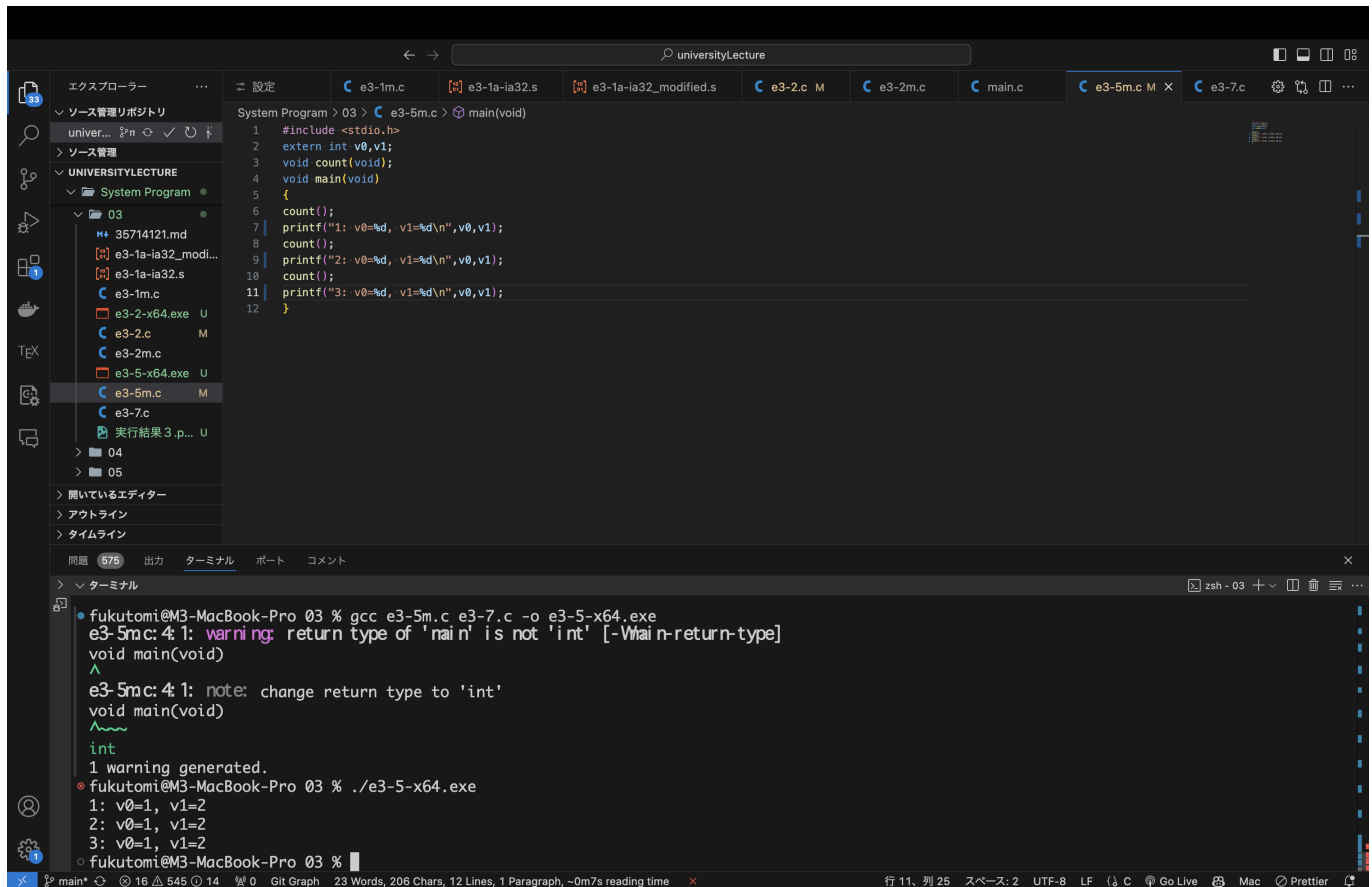
このコードはprintfを使用して、各変数のアドレスとサイズを出力している。

&演算子を用いて変数のアドレスを取得し、sizeof演算子で変数のサイズを取得している。データ型によってサイズが違うので適切に使用することでメモリ使用量やパフォーマンスを向上させることができる。

④ e3-5m.cとe3-7.cの2つのファイルを作成してコンパイルし、実行ファイルを実行した結果を報告してください。

## 実行結果

画像の下部分のターミナルに実行結果あります



The screenshot shows a VS Code editor with a project named 'universityLecture'. The file explorer on the left shows a directory structure with files like 'e3-1a-ia32.s', 'e3-1a-ia32\_modified.s', 'e3-1m.c', 'e3-2-x64.exe', 'e3-2.c', 'e3-2m.c', 'e3-5-x64.exe', 'e3-5m.c', and 'e3-7.c'. The main editor window shows the content of 'e3-5m.c', which is a C program with a 'main' function that calls 'count' three times, each time printing the values of 'v0' and 'v1'. The terminal at the bottom shows the compilation and execution of the program. The compilation command is 'gcc e3-5m.c e3-7.c -o e3-5-x64.exe', which produces a warning about the return type of 'main'. The execution command is './e3-5-x64.exe', which outputs the results of the three 'count' calls: '1: v0=1, v1=2', '2: v0=1, v1=2', and '3: v0=1, v1=2'.

```
System Program > 03 > e3-5m.c > main(void)
1 #include <stdio.h>
2 extern int v0,v1;
3 void count(void);
4 void main(void)
5 {
6     count();
7     printf("1: v0=%d, v1=%d\n",v0,v1);
8     count();
9     printf("2: v0=%d, v1=%d\n",v0,v1);
10    count();
11    printf("3: v0=%d, v1=%d\n",v0,v1);
12 }
```

```
fukutomi@M3-MacBook-Pro 03 % gcc e3-5m.c e3-7.c -o e3-5-x64.exe
e3-5m.c:4:1: warning: return type of 'main' is not 'int' [-Wmain-return-type]
void main(void)
^
e3-5m.c:4:1: note: change return type to 'int'
void main(void)
^
int
1 warning generated.
fukutomi@M3-MacBook-Pro 03 % ./e3-5-x64.exe
1: v0=1, v1=2
2: v0=1, v1=2
3: v0=1, v1=2
fukutomi@M3-MacBook-Pro 03 %
```

## 作成したファイル内容、結果について

e3-7.cでは自動変数でcを定義しているので毎回変数が消滅してしまいカウントできていない。これを防ぐためには、自動変数ではなく大域変数を使うか、自動変数でもカウントできるようにプログラムを作る必要がある。

## 講義に対する感想・質問・意見

様々な変数の型があり、それぞれの特性がかなり違うので、変数を宣言する時はどの型で宣言するのが適切かをしっかり考えるべきだと感じた。

また、様々な型が持つ特性について、なぜそのような特性を持つかを理解することでバグの原因を突き止め安くなると思う。