Embedded computing for scientific and industrial imaging applications

Lecture 17 - Binary vs text output, Course summary

Binary vs. text(ASCII) output

Often need to write out a large array of floats with full precision.

For example, one solution value on 3d grid …

```
for (i=0; i < n; i++)
    for (j=0; j < n; j++)
        for (k=0; k < n; k++)
            fprintf(fp, "%24.16e", u[i][j][k]);
```

How much disk space does this take?

ASCII stands for American Standard Code for Information Interchange

Binary vs. text(ASCII) output

Often need to write out a large array of floats with full precision.

For example, one solution value on 3d grid …

```
for (i=0; i < n; i++)
    for (j=0; j < n; j++)
        for (k=0; k < n; k++)
            fprintf(fp, "%24.16e", u[i][j][k]);
```

How much disk space does this take?

A single number such as

“ `1.0000000000000000e+00` ” has 24 ASCII characters ⇒ 24 bytes per value.

Total $24n^3$ bytes. E.g. 100 × 100 × 100 grid: n = 100 ⇒ 24 MB.

ASCII stands for American Standard Code for Information Interchange

Binary vs. text(ASCII) output

Often need to write out a large array of floats with full precision.

For example, one solution value on 3d grid …

```
for (i=0; i < n; i++)
    for (j=0; j < n; j++)
        for (k=0; k < n; k++)
            fprintf(fp, "%24.16e", u[i][j][k]);
```

How much disk space does this take?

A single number such as

“ `1.0000000000000000e+00` ” has 24 ASCII characters ⇒ 24 bytes per value.

Total $24n^3$ bytes. E.g. 100 × 100 × 100 grid: n = 100 ⇒ 24 MB.

Note: In memory storing one 8-byte float takes only 8 bytes. (n = 100 =⇒ 8MB.) ASCII takes 3× the space.

ASCII stands for American Standard Code for Information Interchange

Binary vs. text(ASCII) output

Often need to write out a large array of floats with full precision.

For example, one solution value on 3d grid …

```
for (i=0; i < n; i++)
    for (j=0; j < n; j++)
        for (k=0; k < n; k++)
            fprintf(fp, "%24.16e", u[i][j][k]);
```

How much disk space does this take?

A single number such as

“ `1.0000000000000000e+00` ” has 24 ASCII characters ⇒ 24 bytes per value.

Total $24n^3$ bytes. E.g. 100 × 100 × 100 grid: n = 100 ⇒ 24 MB.

Note: In memory storing one 8-byte float takes only 8 bytes. (n = 100 =⇒ 8MB.) ASCII takes 3× the space.

Also takes additional time to convert to ASCII, ≃ 10× slower to write ASCII than dumping binary.

ASCII stands for American Standard Code for Information Interchange

Binary output in C

```
$(CSE6000)/codes/17_codes/hello_file.c
void double2bin(char* filepath, double* val, int size)
{
    FILE *fp;
    fp = fopen(filepath, "w");
    fwrite(val, size, sizeof(double), fp);
    fclose(fp);
}
```

The resulting binary file `dummy.bin` cannot be edited directly.
But we can read it into MATLAB...

Reading binary data files in MATLAB

```
$(CSE6000)/codes/17_codes/bin2double.m
function data = bin2double(filename, sizeA)
    fileID = fopen(filename, 'r');
    data = fread(fileID, sizeA, 'double');
    fclose(fileID);
end
```

Check `double2bin.m` for writing binary data.

Summary, take away messages…

- Version control — git, GitHub
  Use for all your projects, collaborations, …
  Consider contributing to open source projects
    Submit a pull request
- C, Visual Studio
  Binary storage, floating point number, C, data type, function, array, pointer
- Computer architecture, cache, optimization
  Memory hierarchy, cache considerations
  Consider layout of arrays in memory Aim for spatial and temporal locality

Summary, take away messages…

- OpenCV, LAPACK
  Affine transformation
- Parallel computing
  Increasingly necessary for all computing
  Amdahl’s law — inherently sequential code limits parallelization
  Weak vs. strong scaling
  Fine grain vs. coarse grain parallelism
- OpenMP
  Assumes shared memory
  Often very easy to add to existing codes
  Need to worry about shared/private variables, race conditions

Happy Holidays & Happy new year
Thanks for participating.