Single file development

Mark Mendell
Handmade NYC Meetup 12/7/2023

```c
int
main(void)
{
    // Do some stuff
    int x = 42;
    ...

    // Other stuff
    printf("hey there");
    int fd = open("foo", O_RDONLY);

    // Cleanup
    close(fd);
    free(ptrs);
    for (int i=0; i<10; i++) free(data[i]);
}
```

```
int
main(void)
{
    DoStuff::MainRoutine();
}
```

```
class DoStuff
{
    public:
    DoStuff(){}
    virtual ~DoStuff(){}
    virtual void MainRoutine() = 0;
}
```

```
class DoStuffReal : DoStuff
{
    private:
    int stuff;

    public:
    Concrete(){}
    virtual ~ForReal(){}
    virtual void MainRoutine() = 0;
}
```

```
void DoStuffReal::MainRoutine()
{
    DoHelper *f = new DoHelper();
    f->Part1();
    f->Part3(f->part2());
}
```

```c
int
main(void)
{
    // Do some stuff
    int x = 42;
    ...

    // Other stuff
    printf("hey there");
    int fd = open("foo", O_RDONLY);

    // Cleanup
    close(fd);
    free(ptrs);
    for (int i=0; i<10; i++) free(data[i]);
}
```

```c
void
dosomestuff(void)
{
    int x = 42;
    ...
}

int
otherstuff(void)
{

    printf("hey there");
    return open("foo", O_RDONLY);
}

void
cleanup(int fd)
{
    close(fd);
    free(ptrs);
    for (int i=0; i<10; i++) free(data[i]);
}

int
main(void)
{
    dosomestuff();
    int fd = otherstuff();
    cleanup();
}
```

- util
- test
- lib
  - serialization
  - parsing
  - cmd
  - util.c
  - runtime.c
  - history.c
  - log.c

tool.c

tool.c

```c
int
main(void)
{
    // Do some stuff
    int x = 42;
    ...

    // Other stuff
    printf("hey there");
    int fd = open("foo", O_RDONLY);

    // Cleanup
    close(fd);
    free(ptrs);
    for (int i=0; i<10; i++) free(data[i]);
}
```

- util
- test
- lib
  - serialization
  - parsing
  - cmd
  - util.c
  - runtime.c
  - history.c
  - log.c

```c
void
dosomestuff(void)
{
    int x = 42;
    ...
}

int
otherstuff(void)
{

    printf("hey there");
    return open("foo", O_RDONLY);
}

void
cleanup(int fd)
{
    close(fd);
    free(ptrs);
    for (int i=0; i<10; i++) free(data[i]);
}

int
main(void)
{
    dosomestuff();
    int fd = otherstuff();
    cleanup();
}
```

MobileCode (Android app)

 mobilecode.c

Common util functions? Util.h? String.h? myarray.h?

- tool.c
- util.h
- string.h
- myarray.h

Copy paste functions in

- tool.c

Problem: cross-platform

Android:

JNI_dev_mark_onCreate
JNI_dev_mark_dispatchTouchEvent

iOS:

applicationDidFinishLaunching
AppDelegate_touchesEnded

Web:

wasmmain
ontouch

Windows:

WinMain
WM_CLICK

Solution(?): #if

```
#if __linux__
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
#elif __APPLE__
void AppDelegate_applicationDidFinishLaunching(id self, SEL _cmd)
#elif __wasm__
void main(void)
#elif _WIN32
int WinMain(HINSTANCE hInstance, HINSTANCE ...)
#endif
{
    // Initialize code
    srand(time(NULL));
    #if _WIN32
    HANDLE hfile = CreateFile("/tmp/log", blah);
    if (!hfile) ...
    #else
    int fdlog = open("/tmp/log");
    if (fdlog == -1) ...
    #endif
}
```

```
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log");
    if (fdlog == -1) ...
}
```

Solution(?): platform layer

- mobilecode.c
- android.h
- ios.h
- wasm.h
- win32.h

```
struct touchevent {
    double x, y;
    unsigned ms;
    char wasrclick;
};

jbool
JNI_dev_mark_dispatchTouchEvent(JNIEnv *env, jobject self, jobject event)
{
    jfloat x=getfieldfjni(event,"x"), y=getfieldfjni(event,"y");
    unsigned ms = getfieldujni(event, "time");
    struct touchevent touchev = {x, y, ms};
    return onclick(touchev);
}

char
onclick(struct touchevent touchev)
{
    if (touchev.x > 10) printf("left tap\n");
}
```

- mobilecode.c

```
jbool
JNI_dev_mark_dispatchTouchEvent(JNIEnv *env, jobject self, jobject event)
{
    jfloat x=getfieldfjni(event,"x"), y=getfieldfjni(event,"y");
    if (x > 10) printf("left tap\n");
}
```

Solution: file per target

mobilecode.c (Android)

```
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log");
    if (fdlog == -1) ...
}

jbool
JNI_dev_mark_dispatchTouchEvent(JNIEnv *env, jobject self,
  jobject event)
{
    jfloat x=getfieldfjni(event,"x"), y=getfieldfjni(event,"y");
    if (x > 10) printf("left tap\n");
}
```

mobilecode.c (iOS)

```
void AppDelegate_applicationDidFinishLaunching(id self, SEL _cmd)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log");
    if (fdlog == -1) ...
}

void
AppDelegate_touchesEnded(id self, SEL _cmd, id touches)
{
    id touch = objc_msgSend(touches, "anyObject");
    CGPoint loc = objc_msgSend(touch, "locationInView", nil);
    if (loc.x > 10) printf("left tap\n");
}
```

Problem: update android, ios out-of-date

mobilecode.c (Android)

```
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log_mobilecode");
    if (fdlog == -1) ...
}
```

mobilecode.c (iOS)

```
void AppDelegate_applicationDidFinishLaunching(id self, SEL _cmd)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log");
    if (fdlog == -1) ...
}
```

Solution: sync and diff

mobilecode.c (Android synced)

```
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log");
    if (fdlog == -1) ...
}
```

mobilecode.c (Android latest)

```
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log_mobilecode");
    if (fdlog == -1) ...
}
```

mobilecode.c (iOS)

```
void AppDelegate_applicationDidFinishLaunching(id self, SEL _cmd)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log");
    if (fdlog == -1) ...
}
```

diff

```
--- android-synced.c    2023-12-07 17:12:44.618293000 -0500
+++ android-latest.c    2023-12-07 17:13:14.883353000 -0500
@@ -2,7 +2,7 @@
 {
     // Initialize code
     srand(time(NULL));
-    int fdlog = open("/tmp/log");
+    int fdlog = open("/tmp/log_mobilecode");
     if (fdlog == -1) ...
 }
```

Solution: sync and diff

mobilecode.c (Android synced)

```
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log_mobilecode");
    if (fdlog == -1) ...
}
```

mobilecode.c (Android latest)

```
void JNI_dev_mark_onCreate(JNIEnv *env, jobject this)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log_mobilecode");
    if (fdlog == -1) ...
}
```

mobilecode.c (iOS)

```
void AppDelegate_applicationDidFinishLaunching(id self, SEL _cmd)
{
    // Initialize code
    srand(time(NULL));
    int fdlog = open("/tmp/log_mobilecode");
    if (fdlog == -1) ...
}
```

diff

```
--- android-synced.c    2023-12-07 17:12:44.618293000 -0500
+++ android-latest.c    2023-12-07 17:13:14.883353000 -0500
@@ -2,7 +2,7 @@
 {
     // Initialize code
     srand(time(NULL));
-    int fdlog = open("/tmp/log");
+    int fdlog = open("/tmp/log_mobilecode");
     if (fdlog == -1) ...
 }
```

Problem: this sucks

- Added a webview? Yeah, you should manually convert that logic
- 99% of time: tweaked a line of shared identical code

mobilecode.c (Android)

```
char*
addsuffix(char *s)
{
    return sprintfs("%s.%s", suffixfile);
}
```

mobilecode.c (iOS)

```
char*
addsuffix(char *s)
{
    return sprintfs("%s.%s", suffixfile);
}
```

Problem: this sucks

- Added a webview? Yeah, you should manually convert that logic
- 99% of time: tweaked a line of shared identical code

mobilecode.c (Android synced)

```
char*
addsuffix(char *s)
{
    return sprintfs("%s.%s", suffixfile);
}
```

mobilecode.c (Android latest)

```
char*
addsuffix(char *s)
{
    return sprintfs("%s%s", suffixfile);
}
```

mobilecode.c (iOS)

```
char*
addsuffix(char *s)
{
    return sprintfs("%s.%s", suffixfile);
}
```

diff

```
--- android-synced.c    2023-12-07 17:12:44.618293000 -0500
+++ android-latest.c    2023-12-07 17:13:14.883353000 -0500
@@ -2,7 +2,7 @@
 char*
 addsuffix(char *s)
 {
-    return sprintfs("%s.%s", suffixfile);
+    return sprintfs("%s%s", suffixfile);
 }
```

Solution?: auto sync when context is the same

Android

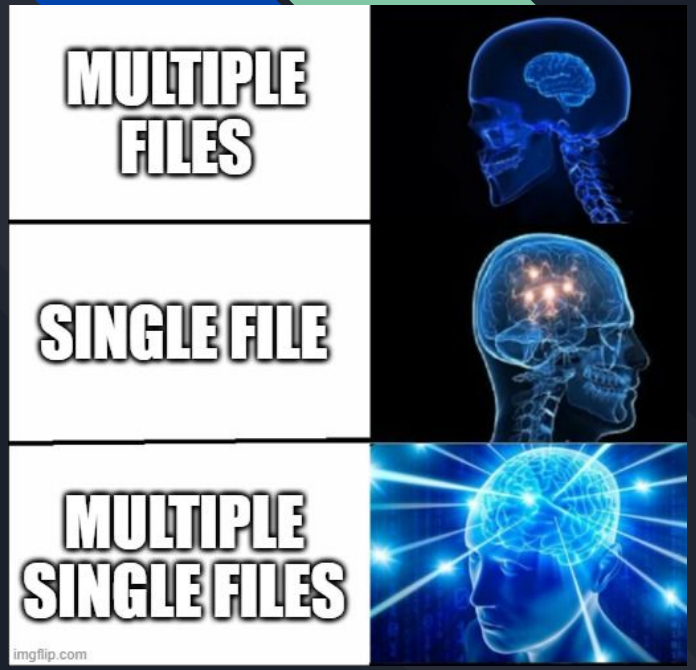
```
char*
addsuffix(char *s)
{
    return sprintfs("%s.%s", suffixfile);
}
```

```
char*
addsuffix(char *s)
{
    return sprintfs("%s%s", suffixfile);
}
```

iOS

```
char*
addsuffix(char *s)
{
    return sprintfs("%s.%s", suffixfile);
}
```

????????



Single file development

Mark Mendell
Handmade NYC Meetup 12/7/2023