CSE306
SOFTWARE QUALITY IN PRACTICE

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall

www.cse.buffalo.edu/faculty/alphonce/SP24/CSE306
MEMORY ORGANIZATION

Each process (a running program) has a chunk of memory at its disposal.

This memory is divided into "static" memory (allocated/structured before execution begins) and "dynamic" memory (allocated while the program executes).
The static segment is divided into a TEXT segment (holding the machine language instructions of the program), a DATA segment (for initialized variables), and a BSS segment (for uninitialized but implicitly zero-assigned values).
The dynamic segment is divided into STACK and a HEAP areas.

The HEAP is generally located adjacent to the STATIC segment, and grows "up" (to higher memory addresses).
The STACK holds invocation records (also called stack frames).

An invocation record is created whenever a function is called. It has space for the function’s parameters, local variables, any return value, as well as bookkeeping information related to the call itself (e.g. where to return to).
TOPHAT QUESTION
git

- distributed version control system
Local Machine
(e.g. your laptop, or cerf if you've ssh'ed in)

- stash
- workspace
- index staging
- local repository

Remote
(e.g. bitbucket, github, CSE servers)

- remote repository
What you see when working

- stash
- workspace
- index staging
- local repository
- remote repository
Cloning a remote

Makes a copy of remote repo in local repo and checks out branch into workspace

stash  workspace  index staging  local repository  remote repository

git clone
Add a file to the staging area (add it to the index)

- stash
- workspace
- index staging
- local repository
- remote repository

`git add`
Create a new commit object with the staged items from the index

Stash → Workspace → Index → Staging → Local Repository → Remote Repository

git commit
Push files from local repo to remote repo

- stash
- workspace
- index staging
- local repository
- remote repository

git push
"git pull is shorthand for git fetch followed by git merge FETCH_HEAD"
[https://git-scm.com/docs/git-pull]
Grab files from remote

- stash
- workspace
- index staging
- local repository
- remote repository

`git fetch`
Create a commit combining the contents of two branches

stash → workspace → index → staging → local repository → remote repository

`git merge`
Let's start by cloning an existing repository
GitIntro

A first repo to show students how to interact with repo on GitHub.
Clone the repo

```bash
% git clone git@github.com:UB-CSE306/git-intro-carl-alphonce-1.git
Cloning into 'git-intro-carl-alphonce-1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 2 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

% ls -la
total 0
drwxr-xr-x  3 alphonce  staff   96 Feb  4 18:10 .
drwxr-xr-x 19 alphonce  staff  608 Feb  4 18:09 ..
drwxr-xr-x  4 alphonce  staff  128 Feb  4 18:10 git-intro-carl-alphonce-1
```
% cd git-intro-carl-alphonce-1
% ls -la

```
total 8
-rw-r--r--   1 alphonce  staff   77 Feb  4 18:10  README.md
drwxr-xr-x  12 alphonce  staff  384 Feb  4 18:10  .git
drwxr-xr-x   3 alphonce  staff   96 Feb  4 18:10  ..
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10  .
```

```bash
% cd .git
% ls -la

total 40
drwxr-xr-x  12 alphonce  staff  384 Feb  4 18:10 .
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10..
-rw-r--r--   1 alphonce  staff   21 Feb  4 18:10 HEAD
-rw-r--r--   1 alphonce  staff  320 Feb  4 18:10 config
-rw-r--r--   1 alphonce  staff   73 Feb  4 18:10 description
drwxr-xr-x  15 alphonce  staff  480 Feb  4 18:10 hooks
-rw-r--r--   1 alphonce  staff  137 Feb  4 18:10 index
drwxr-xr-x   3 alphonce  staff   96 Feb  4 18:10 info
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10 logs
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10 objects
-rw-r--r--   1 alphonce  staff  112 Feb  4 18:10 packed-refs
drwxr-xr-x   5 alphonce  staff  160 Feb  4 18:10 refs
```
pointer to the current branch

-rw-r--r--  1 alphonce  staff  21 Feb  4 18:10 HEAD
-rw-r--r--  1 alphonce  staff  320 Feb  4 18:10 config
-rw-r--r--  1 alphonce  staff   73 Feb  4 18:10 description
drwxr-xr-x  15 alphonce  staff  480 Feb  4 18:10 hooks
-rw-r--r--  1 alphonce  staff  137 Feb  4 18:10 index
drwxr-xr-x   3 alphonce  staff   96 Feb  4 18:10 info
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10 logs
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10 objects
-rw-r--r--  1 alphonce  staff  112 Feb  4 18:10 packed-refs
drwxr-xr-x   5 alphonce  staff  160 Feb  4 18:10 refs
<table>
<thead>
<tr>
<th>Mode</th>
<th>Owner</th>
<th>Group</th>
<th>Size</th>
<th>Date</th>
<th>Time</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>21</td>
<td>Feb 4</td>
<td>18:10</td>
<td>HEAD</td>
</tr>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>320</td>
<td>Feb 4</td>
<td>18:10</td>
<td>config</td>
</tr>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>73</td>
<td>Feb 4</td>
<td>18:10</td>
<td>description</td>
</tr>
<tr>
<td>drwx</td>
<td>alphonce</td>
<td>staff</td>
<td>480</td>
<td>Feb 4</td>
<td>18:10</td>
<td>hooks</td>
</tr>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>137</td>
<td>Feb 4</td>
<td>18:10</td>
<td>index</td>
</tr>
<tr>
<td>drwx</td>
<td>alphonce</td>
<td>staff</td>
<td>96</td>
<td>Feb 4</td>
<td>18:10</td>
<td>info</td>
</tr>
<tr>
<td>drwx</td>
<td>alphonce</td>
<td>staff</td>
<td>128</td>
<td>Feb 4</td>
<td>18:10</td>
<td>logs</td>
</tr>
<tr>
<td>drwx</td>
<td>alphonce</td>
<td>staff</td>
<td>128</td>
<td>Feb 4</td>
<td>18:10</td>
<td>objects</td>
</tr>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>112</td>
<td>Feb 4</td>
<td>18:10</td>
<td>packed-refs</td>
</tr>
<tr>
<td>drwx</td>
<td>alphonce</td>
<td>staff</td>
<td>160</td>
<td>Feb 4</td>
<td>18:10</td>
<td>refs</td>
</tr>
</tbody>
</table>
staging area

-rw-r--r--  1 alphonce  staff  21 Feb  4 18:10 HEAD
-rw-r--r--  1 alphonce  staff  320 Feb  4 18:10 config
-rw-r--r--  1 alphonce  staff   73 Feb  4 18:10 description
drwxr-xr-x  15 alphonce  staff  480 Feb  4 18:10 hooks
-rw-r--r--  1 alphonce  staff  137 Feb  4 18:10 index
drwxr-xr-x   3 alphonce  staff   96 Feb  4 18:10 info
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10 logs
drwxr-xr-x   4 alphonce  staff  128 Feb  4 18:10 objects
-rw-r--r--  1 alphonce  staff  112 Feb  4 18:10 packed-refs
drwxr-xr-x   5 alphonce  staff  160 Feb  4 18:10 refs
content
(blobs, trees, commits)
The *git man page* seems to be surprisingly bereft of an official definition, other than this (emphasis mine):

The **object database** contains objects of three main types: **blobs**, which **hold file data**; trees, which point to blobs and other trees to build up directory hierarchies; and commits, which each reference a single tree and some number of parent commits.

The repeated use of the term "object database" across git documentation suggests a borrowing of "blob" specifically from DBMSs.

In its article on [Binary large objects](https://en.wikipedia.org/wiki/Binary_large_objects) Wikipedia defines the term as "a collection of binary data stored as a single entity in a database management system", further offering the following:

Blobs were originally just amorphous chunks of data invented by Jim Starkey at DEC, who describes them as "the thing that ate Cincinnati, Cleveland, or whatever" from "the 1958 Steve McQueen movie", referring to The Blob. Later, Terry McKiever, a marketing person for Apollo, felt that it needed to be an acronym and invented the backronym Basic Large Object. Then Informix invented an alternative backronym, Binary Large Object.

So, though it's not a definitive answer, the term "blob" has a conventional and well-defined usage across computer science as an opaque string of binary data, and git adheres to that definition without further specifying it.
Initial commit%
Initial commit:

```
alphonce@dhcp-10-83-16-71 .git % git cat-file -t e7c77
tree
```

```
alphonce@dhcp-10-83-16-71 .git % git cat-file -p e7c77
100644 blob da80e272f601badf16da372517a26ba132b5bab
100644 blob db80e272f601badf16da372517a26ba132b5bab
```

```
# GitIntro
A first repo to show students how to interact with repo on GitHub
```
NAME

git-cat-file - Provide contents or details of repository objects

SYNOPSIS

```
git cat-file <type> <object>
git cat-file (-e | -p) <object>
git cat-file (-t | -s) [--allow-unknown-type] <object>
git cat-file (--textconv | --filters)
   [rev:<path|tree-ish> | --path=<path|tree-ish> <rev>]
git cat-file (--batch | --batch-check | --batch-command) [--batch-all-objects]
   [--buffer] [--follow-symlinks] [--unordered]
   [--textconv | --filters] [-z]
```

DESCRIPTION

Output the contents or other properties such as size, type or delta information of one or more objects.

This command can operate in two modes, depending on whether an option from the `--batch` family is specified.

In non-batch mode, the command provides information on an object named on the command line.

In batch mode, arguments are read from standard input.

OPTIONS

- `<object>`

   The name of the object to show. For a more complete list of ways to spell object names, see the
   "SPECIFYING REVISIONS" section in `gitrevisions(7)`.

- `-t`

   Instead of the content, show the object type identified by `<object>`.

- `-s`

   Instead of the content, show the object size identified by `<object>`. If used with `--use-mailmap` option, will show the size of updated object after replacing idents using the mailmap mechanism.

- `-e`

   Exit with zero status if `<object>` exists and is a valid object. If `<object>` is of an invalid format, exit with non-zero status and emit an error on stderr.

- `-p`

   Pretty-print the contents of `<object>` based on its type.
# GitIntro

A first repo to show students how to interact with repo on GitHub
Could even do:

```bash
 alphonce@dhcp-10-83-16-71 .git % git cat-file -t HEAD
commit
alphonce@dhcp-10-83-16-71 .git % git cat-file -t refs/heads/main
commit
```

```
Initial commit
alphonce@dhcp-10-83-16-71 .git % git cat-file -t e7c77
tree
alphonce@dhcp-10-83-16-71 .git % git cat-file -p e7c77
100644 blob da80e272f601badf16da372517a26bba132b5bab README.md
alphonce@dhcp-10-83-16-71 .git % git cat-file -t da80e
blob
alphonce@dhcp-10-83-16-71 .git % git cat-file -p da80e
# GitIntro
A first repo to show students how to interact with repo on GitHub
```
Initial commit

A first repo to show students how to interact with repo on GitHub
Initial commit

```
alphonce@dhcp-10-83-16-71 .git % git cat-file -t e7c77
tree e7c77e150f84d16ec763bd46dcffebebf8f2873f4
author github-classroom[bot] <66690702+github-classroom[bot]@users.noreply.github.com> 1707088127 +0000
committer GitHub <noreply@github.com> 1707088127 +0000
gpgsig -----BEGIN PGP SIGNATURE-----
wsFcBAABCAAQBQJlwBj/CRC1aQ7uu5UhlAAAbt8QAJvL5XVgE7xXxNDM+ZqUWQgp
nAdFnL03n0fMqHskU9hG8FFnSw3DODZinpZ4se7cwqFdVoDoH8ePCXDrZHKkmDq5c
6jwkejReAuFwyaYggqu5j1YSdnPnPpPFWjSmHZHawf6TIN0D+VZn4YhjJcmLdpq
ij4WmeB4oGqcEUVr0m11mgcZSeCusrQV4yKTvKtwzJoh6Z5UonYlk0aNxD6dGZ/9
lUDr0/2NrpqUmoR2w1v9+0F334LLK520y2CDV3DfKvoDqDM80TD6hStZBByImn
ji0wxBhbh4H2o3OH/eqvV850GtAvarykKa1WfhtKkoBKEY77u9F63biuFbeXyjY
UrT2AgFHoVlaOttrtfx0Kad5FJyux6YXXX/gbPO/uu4Dlx1i4wQdcDAT9VGhkJ
4Cycu5g0Zfi5x08tC8V78Ov00W0Vhok9+SowoD+pJ5E14qsBs9MYxenYH6cc45
+Dh4Eq14bdj77TkTALGaaX1Hy25p+R0Cm/0/11lUol6g/77mPyCnZMYB6+F0oPa7q
NBDnsphM00oX7Ms1gko3HeXw8sNJUbslnZq9OPPp9d83dQVxKfwa+LX1+L3
wwh5jNSIepeA/3BiPAdcgFsvkJpZ5qleZQe6AAGTR5rUNZtm2xy7UEqLVAjucpREZ
PHhEGRBBD1EWHvQArh
=onC9
-----END PGP SIGNATURE-----
```

A first repo to show students how to interact with repo on GitHub
A first repo to show students how to interact with repo on GitHub
alphonce@dhcp-10-83-16-71 .git % cat HEAD
ref: refs/heads/main
alphonce@dhcp-10-83-16-71 .git % cat refs/heads/main
2303f4622cb330c9fb021cd216cd657e4942592
alphonce@dhcp-10-83-16-71 .git % git cat-file -t HEAD
commit
alphonce@dhcp-10-83-16-71 .git % git cat-file -t 2303f
commit
alphonce@dhcp-10-83-16-71 .git % git cat-file -p 2303f
tree e7c77e150f84d16ec763bd46dcffe8bd8f82873f
author github-classroom[bot] <66690702+github-classroom[bot]@users.noreply.github.com> 1707088127 +0000
committer GitHub <noreply@github.com> 1707088127 +0000
gpgsig -----BEGIN PGP SIGNATURE-----
wsFcBAABCAAQBQJlwBAAC1Q7uu5UhIAAAbt8QAJvL5XVgE7xXxNDM+ZqUWQgp
nAdFnl03n0FMqHskU9g8FFnSw3DDDZinpZ4se7cwqFdvDoH8ePcXDrZHkpMqDq5c
o6jwikHzRwFwvaYggu5jn1ysdpnpPPFWjSmH7QawF6TNI0d+V2n4YjhjJcmLdpq
iJ4WmeB4oGqCUEVr0m11mgcZSeCusrQV4yKTvKtwZjOh6Z5UonYlk0aNxdD6gZ9/
LTUDr0/2nrpqRmoRw2vV9+0F334LLK51Ony2CVDV3IKvodDDM80TD6hStZBBYism
ji0wxHbbbh4H2o3QH/eqvV850GTavaryaK19WfhtKkoBKEF777u9F63biuFbeXjY
URTr2TAgFh0ViatwtrFxo0Kad5F3yux6YXXX/npP0/uu4DIxl14wQdCDA9VGHkJ
44CucSgG0Zfa15x08tC88V80mwXXVHok9+SwooD+pJ5E14qsBs9MYxenYH6cc45
+Dh4gEeq4bdJ77TkTAIgaaXHy2Sp+R0Cn0/iUul6g77tmPcyCMZ8YB6+Fo0pa7q
NBGDnsphM00kX7Ms1Gki03HeXw8SNJubsnlcZg9OPP7p9d93dQkxWwa4Lx+L3
wwh5jNYepteA/3BiPACdFsvkJpZ5qleZQe6ATR5rUNZtm2xy7UEqLAVjucpREZ
PHhEGRBBD1EHtvQlArh
=onC9
-----END PGP SIGNATURE-----

Initial commit
alphonce@dhcp-10-83-16-71 .git % git cat-file -t e7c77
tree
alphonce@dhcp-10-83-16-71 .git % git cat-file -p e7c77
100644 blob da80eb272f601badf16da372517a26bba132b5bab README.md
alphonce@dhcp-10-83-16-71 .git % git cat-file -t da80e
blob
alphonce@dhcp-10-83-16-71 .git % git cat-file -p da80e
# GitIntro
A first repo to show students how to interact with repo on GitHub