# (ab)using

# GitHub Actions

## for building and testing kernels

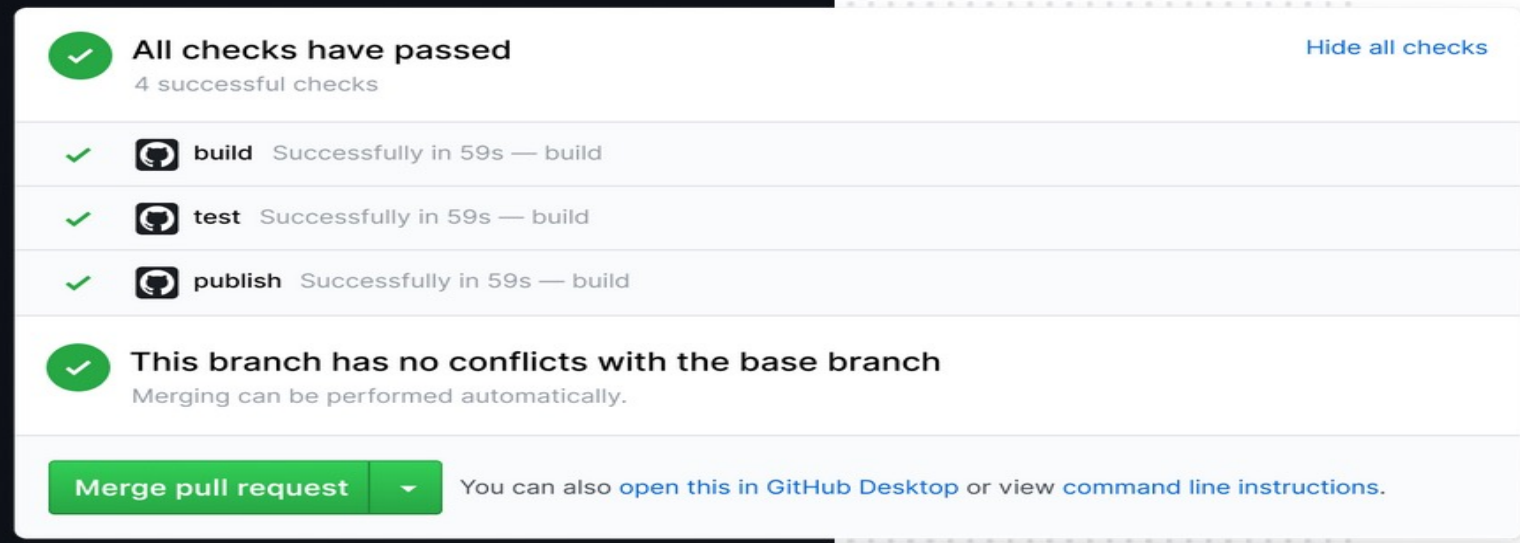
Russel Currey

LINUXCONFAU

GitHub Actions

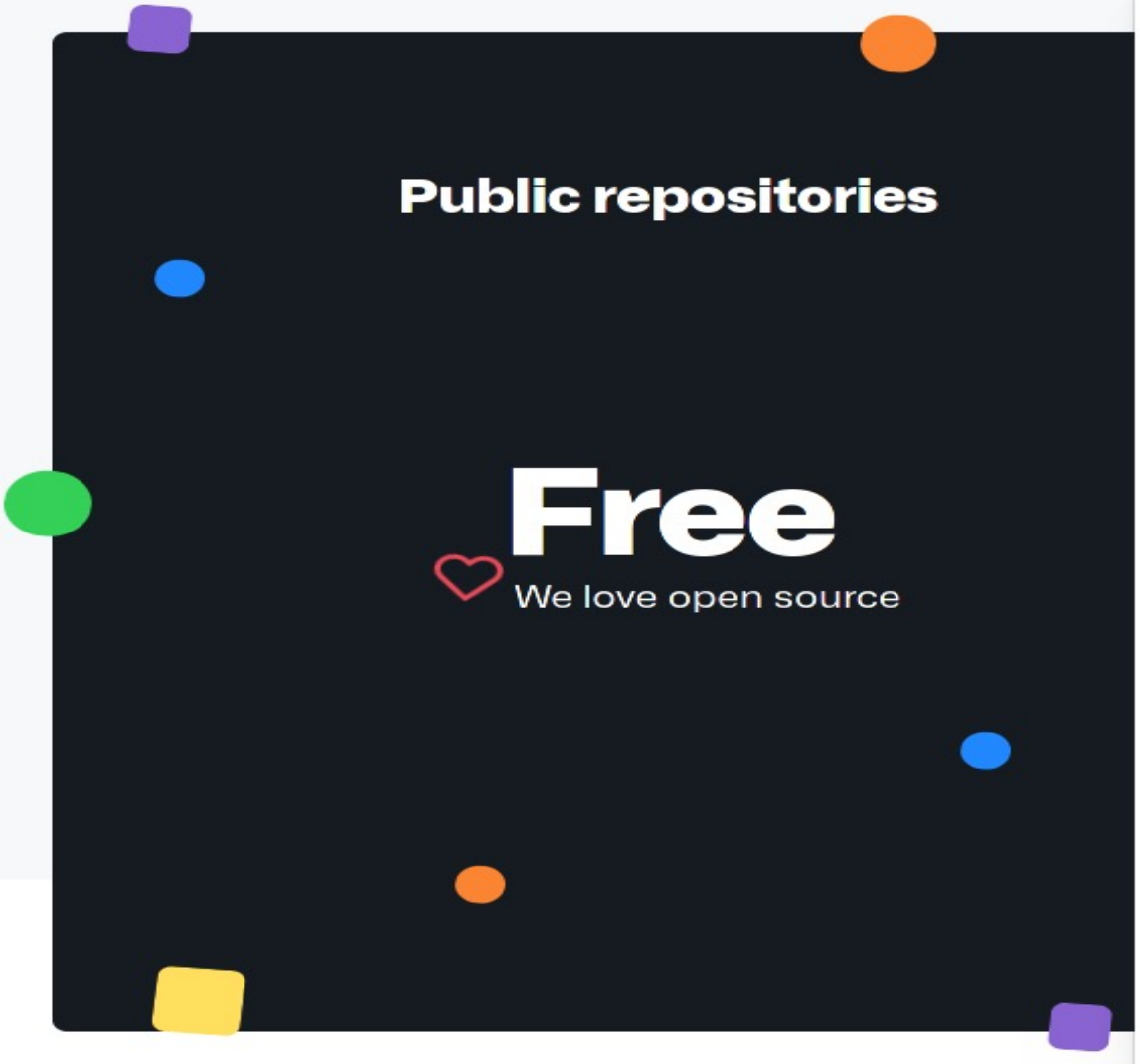
# Automate your workflow from idea to production

```
    ✓ should respond with users 2 and 3 as json

route-map
  GET /users
    ✓ should respond with users
  DELETE /users
    ✓ should delete users
  GET /users/:id
    ✓ should get a user
  GET /users/:id/pets
    ✓ should get a users pets
  GET /users/:id/pets/:pid
    ✓ should get a users pet

route-separation
  GET /
    ✓ should respond with index
  GET /users
    ✓ should list users
  GET /user/:id
    ✓ should get a user
    ✓ should 404 on missing user
```




unlimited
free
compute

## Public repositories




## Private repositories

| Included minutes | | Additiona |
|---|---|---|
| Free | 2,000 minutes per month | Linux 2 cores, 7C |
| Pro | 3,000 minutes per month | Windows 2 cores, 7C |
| Team | 3,000 minutes per month | macOS 2 cores, 7C |
| Enterprise | 50,000 minutes per month | Self-hoste |

Included, hosted runner minutes are consumed at different ra
GitHub Actions is not available for private repos in legacy per-

# what can it do?

# what can it do?

anything you can think of…

# what can it do?

anything you can think of…

…that runs inside a low performance VM

# let's build a kernel

we need:

- GitHub repository
- Kernel source
- Workflow file

# what's a workflow file?

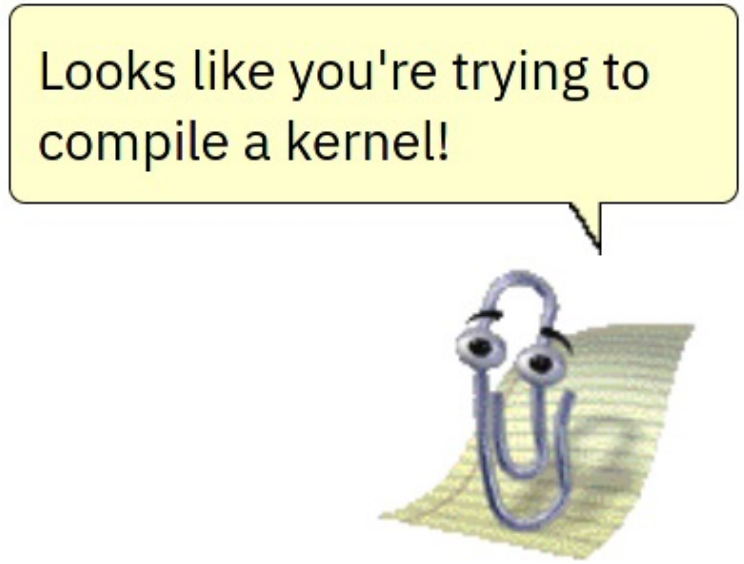
- YAML format
- What triggers the action?
- What does the action run on?
- What does the action actually do?

# triggers

- Push (specific branch? any branch)
- Pull request
- Manual
- Commit filters
  - only test a given subsystem if it's actually changed

# hosted runners

- Latest Ubuntu LTS (20.04)
- Previous Ubuntu LTS (18.04)
- macOS and Windows if you’re into that
- Azure Dv2/DSv2 instance
- 2-core CPU (Haswell at the oldest)
- 7GB RAM
- 14GB SSD



# kernel build, part 1

```
name: kernel_build
on:
  push
jobs:
  kernel:
     runs-on: ubuntu-latest
```

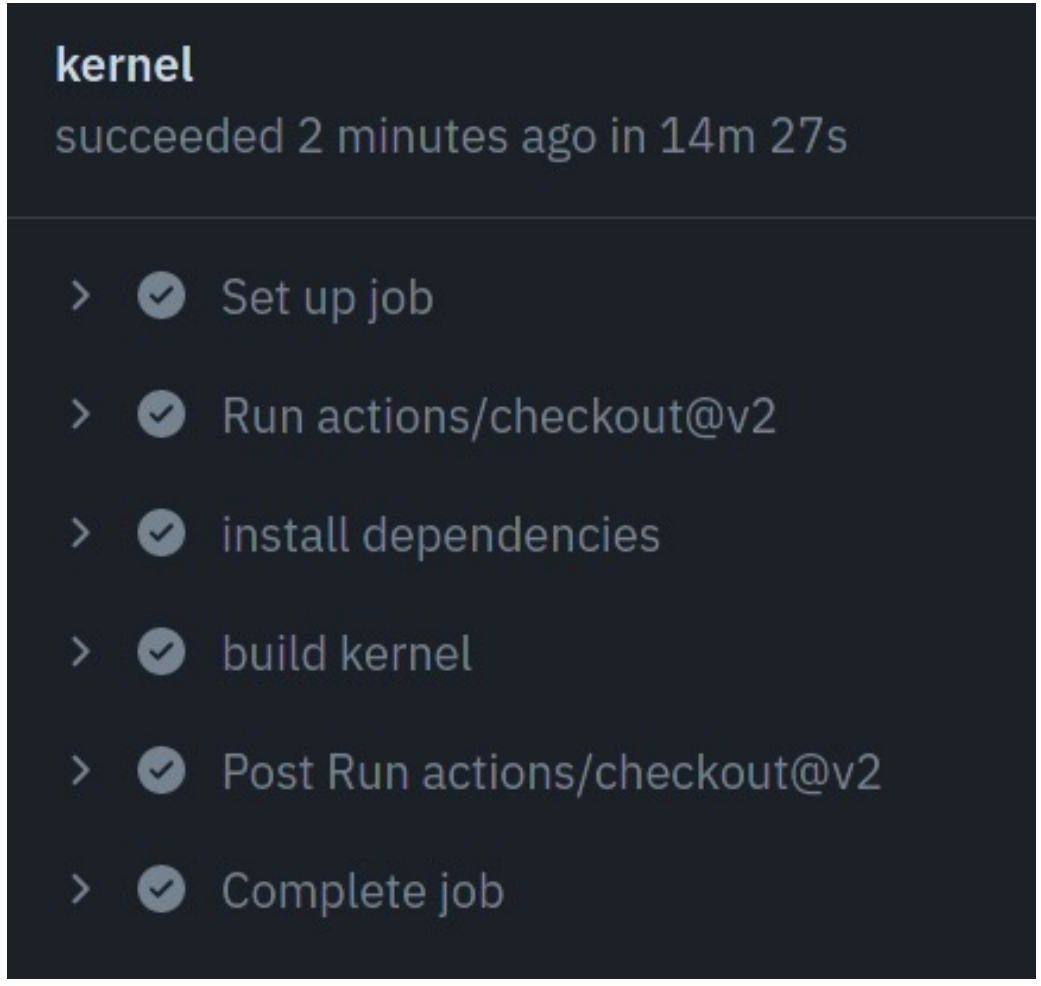
# kernel build, part 2

```
steps:
- uses: actions/checkout@v2
- name: install dependencies
  run: sudo apt install -y gcc make …
- name: build kernel
  run: make defconfig && make -j`nproc`
```

# thanks, someone else's computer!

we ran make, slowly.

what else can we do?



# enter the matrix

```
strategy:
  matrix:
     defconfig: [defconfig, allyesconfig,
                 allmodconfig, allnoconfig]
…
run: make ${{ matrix.defconfig }} && make
```

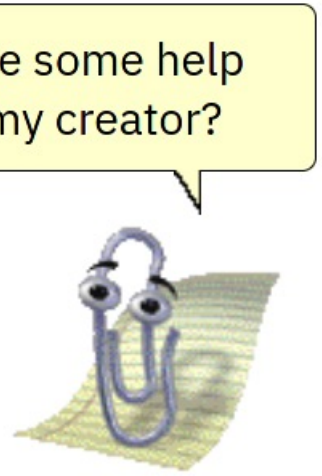
# four times the power

Triggered via push 17 minutes ago

ruscur pushed 8182c3a `lca`

| Status | Total duration |
| --- | --- |
| **Success** | **17m 4s** |

- kernel (defconfig)
- kernel (allyesconfig)
- kernel (allmodconfig)
- kernel (allnoconfig)



# don't like ubuntu?

- first-class Docker support
- community helpers for lxd, podman etc

# ccache

- GitHub Actions cache feature lets you preserve state

```
- name: Load ccache
  uses: actions/cache@v2
  with:
    path: ~/.ccache
    key: ${{ matrix.defconfig }}
```

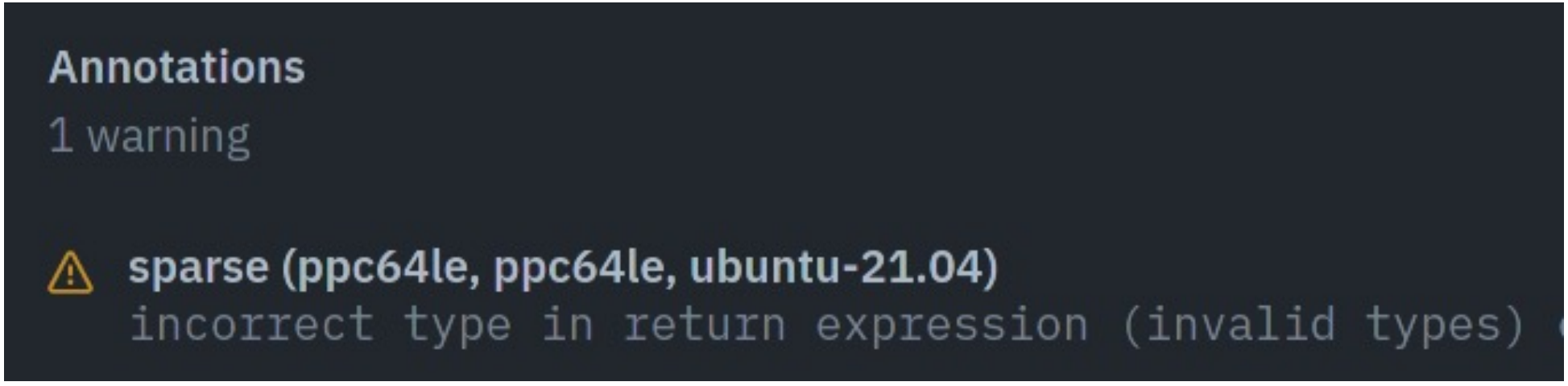
# problem matchers

- Teach GitHub Actions what your output looks like

```
"problemMatcher": [
        {
            "owner": "powerpc-sparse",
            "pattern": [
                {"regexp": "^(?:\\+|-
)?(?:/linux/)?(.*):(\\d+):(\\d+):\\s+(error|wa
rning):\\s+(.*)$", …
```

# thanks, regex!

+/linux/arch/powerpc/kernel/module_64.c:57:16: warning:
incorrect type in return expression (invalid types)
expected struct func_desc_t got struct func_desc



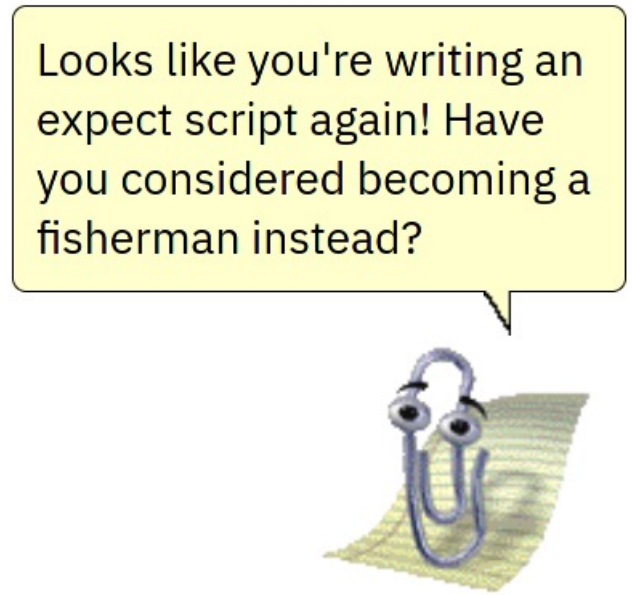

```
- name: Archive artifacts
  uses: actions/upload-artifact@v2
  with:
    name: build-${{ matrix.defconfig }}.log
    path: ~/linux/build.log
```

# check for regressions

```
- name: Get results from upstream build
  uses: dawidd6/action-download-artifact@v2
  with:
    workflow: kernel-build.yml
    workflow_conclusion: success
    branch: master
    name: build-${{ matrix.defconfig }}.log
```

# booting in qemu

- take your vmlinux
- run qemu
- use *expect* to make it do stuff
- check if it does the stuff
- cry (profusely)



# putting it all together

- GitHub Actions runs some CI for arch/powerpc
  - kernel development mailing list
  - GitHub repository with actions in it
  - Patchwork server
    - API for acquiring patch series
    - API to publish test results to
  - snowpatch

# Patchwork

Patchwork Linux PPC development | Patches | Bundles | About this project

Show patches with: State = **Action Required** | Archived = **No** | 720 patches

| Patch | Series | A/F/R/T | S/W/F |
|---|---|---|---|
| powerpc/audit: Fix syscall_get_arch() | powerpc/audit: Fix syscall_get_arch() | - 1 - - | 5 - - |
| [v3] powerpc/papr_scm: Implement initial support for injecting smart errors | [v3] powerpc/papr_scm: Implement initial support for injecting smart errors | - - - - | 5 - - |
| [v2,2/2] powerpc/fadump: opt out from freeing pages on cma activation failure | powerpc/fadump: handle CMA activation failure appropriately | 1 - - - | 5 - - |
| [v2,1/2] mm/cma: provide option to opt out from exposing pages on activation failure | powerpc/fadump: handle CMA activation failure appropriately | - - 1 - | - - - |
| [1/1] powerpc/e500/qemu-e500: allow core to idle without waiting | [1/1] powerpc/e500/qemu-e500: allow core to idle without waiting | - - - - | 5 - - |
| [v2] powerpc: dts: t1040rdb: fix ports names for Seville Ethernet switch | [v2] powerpc: dts: t1040rdb: fix ports names for Seville Ethernet switch | - 1 2 - | 5 - - |
| powerpc: dts: add device tree for T1040RDB-REV-A | powerpc: dts: add device tree for T1040RDB-REV-A | - - - - | 5 - - |
| [v5,6/6] module: Move duplicate mod_check_sig users code to mod_parse_sig | KEXEC_SIG with appended signature | - - - - | 5 - - |

# snowpatch

- watches Patchwork for new series
- applies them to a local git tree
- pushes a new branch to GitHub
- waits for Actions to complete
- sends results back to Patchwork

# arch/powerpc workflows

- big matrix of 32/64bit, BE/LE, platform, defconfig
- builds & boots in qemu
- gcc & clang
- *sparse* diffs between maintainer's tree and new patches
  - problem matcher registered on that diff for regressions
- mostly written by Michael Ellerman

# end result

## [v3,12/12] lkdtm: Add a test for function descriptors protection

| | |
|---|---|
| **Message ID** | 67f9545c9ad15048bfe0104278ef9595d051dbc8.1634457599.git.christophe.leroy@csgroup.eu |
| **State** | New |
| **Headers** | show |
| **Series** | Fix LKDTM for PPC64/IA64/PARISC \| expand |
| **Related** | show |

### Checks

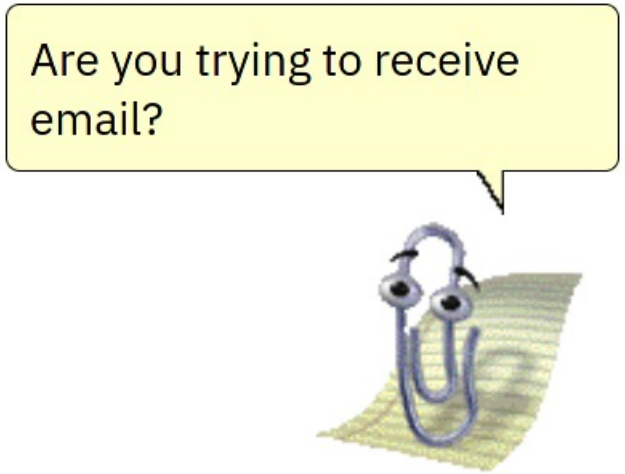
| Context | Check | Description |
|---|---|---|
| snowpatch_ozlabs/github-powerpc_selftests | **success** | Successfully ran 8 jobs. |
| snowpatch_ozlabs/github-powerpc_ppctests | **success** | Successfully ran 8 jobs. |
| snowpatch_ozlabs/github-powerpc_sparse | **warning** | sparse (ppc64le, ppc64le, ubuntu-21.04) found 1 issues. |
| snowpatch_ozlabs/github-powerpc_clang | **success** | Successfully ran 7 jobs. |
| snowpatch_ozlabs/github-powerpc_kernel_qemu | **success** | Successfully ran 24 jobs. |

# potential use cases

- helpful for subsystem maintainers
- developer pre-submit checks
- helping new developers get started
- automating anything with a big matrix

# limitations

- spamming too many jobs at once can get you rate limited
- no guarantee you'll get capacity when you want it
- occasional hiccups that make your jobs fail
- potential for email DDOS

# that's all folks

- Questions?
- thanks to all linux.conf.au organisers
- @russelldotcc / ruscur@russell.cc